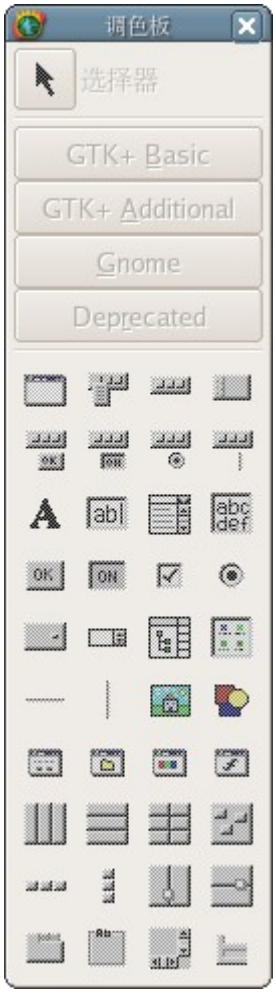
Glade 编程快餐 v1.02

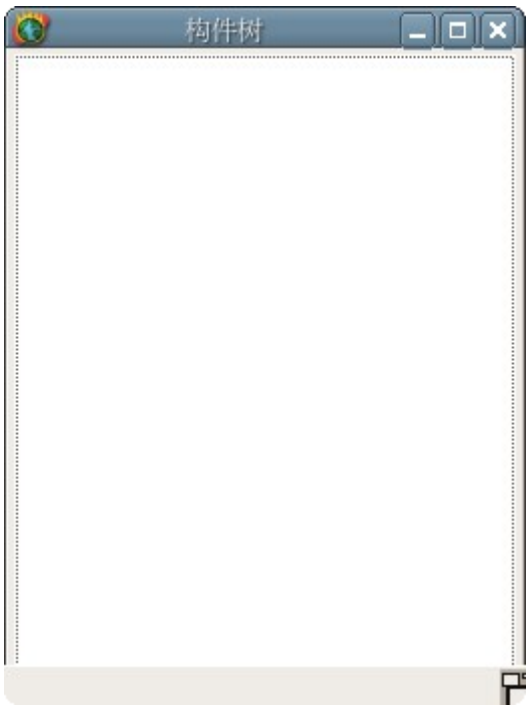
开发基于 GTK+库的管网平差程序







Glade makes you glad!



目录

简介……3
一、启动 glade2.10.0 ……4
二、构造图形界面……4
三、保存……7
四、修改 ……10
五、C 代码联编和生成 ……13
六、修改 callbacks.c 源代码来适应需要的功能……14
七、美化 ……15
八、GTK+的工作机理 ……21
九、附录（管网平差源代码）……23

简介

福利

您熟悉 C 语言吗？您想编写图形界面程序吗？

如果回答是，那不需要另外的理由，您完全可以也应该阅读下面的内容——相当精彩、简单、高效！

开发图形界面通常都需要工具，我们要用的工具不是 Delphi、Visual Basic 或 Power Builder，而是 glade，它基于 Linux 平台（这里介绍的是 RedHat 公司的 Fedora Core 4 发行版）GTK+2.x，由 GNU 的 GCC 编译器进行编译，所有这些基本条件都是免费的（可以从网上免费下载）或用少量的钱 50~60 元购买一套光盘，当然我们喜欢它不仅是因为免费，更因为它自由、高效、稳定，就象我们喜欢 Linux 一样。

Glade 是什么？封面是 Glade 运行时的样子，它是专门为构建 GTK+库基础上的图形界面而设计的工具，您可以去 http://glade.pn.org 去了解更多信息。简单说它是一套可以为你联编代码的图形界面构造器，有了它，您就只要关心怎样象摆积木一样组装图形界面，甚至不会 C 语言也能生成由 C 语言组成的图形界面，当然空架子没有用，你还须用 C 语言来完成具体的功能模块……看看下面的实例，您就能了解它是如何的简单、神奇、激动人心！！！

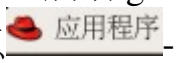
GTK+是什么？它是 GIMP Toolkit 的缩写，是一个开发图形界面的库，您可以去 http://www.gtk.org 获取更多的信息，GTK+是一个跨平台的库，因此即使对于 MSWindows 的用户，也会很熟悉它。GTK+有一个非常独特的特点：它是由 C 语言来实现面向对象的编程的，所以用它来进行图形界面编程，您无须学习 C++而只要 C 语言基础就够了。

介绍编程总得有例子，我们的实例是市政给水专业的管网平差计算程序——一个真实实用的例子，如果你是给排水工程专业，那你就一举两得了！不过请您不要太介意编写的作者的履历——我，还只是一个仅有过一个自建项目的 DOS 下 C 语言编程的初级非专业选手 ；而且还不是计算机专业，只在大学里受了一点计算机教育 ；对于 glade 和 GTK+也是第一次尝试，因此新鲜感和不成熟是在所难免的，如有不当，就权当抛砖引玉吧！不过在另外一个角度讲，这个实例是小弟一直以来自许的精品，功能还是相当强大的，而且，新手与新手更有共同语言，如果你是一个新手的话，那本文是最适合你不过了！

本文是用 OpenOffice 2.0 制作的 PDF 文档，如果您觉得还行，以后也不妨推荐朋友一起来使用，那真是相当的好！重要的是它又好又免费，如果你的荷包还不大鼓的话，不用还等什么呢?

本文以 GPL(GNU General Public Licence 中文译作 GNU 通用公共许可)版权发布，Linux、GCC 等软件都是以之发布的，您可以依之随意复制、发布。

一、启动 glade2.10.0

点

-->编程->Glacde Interface Designer

出现 P.1 所示界面。

二、构造图形界面



本图是主界面，点新建可以建立一个新的项目

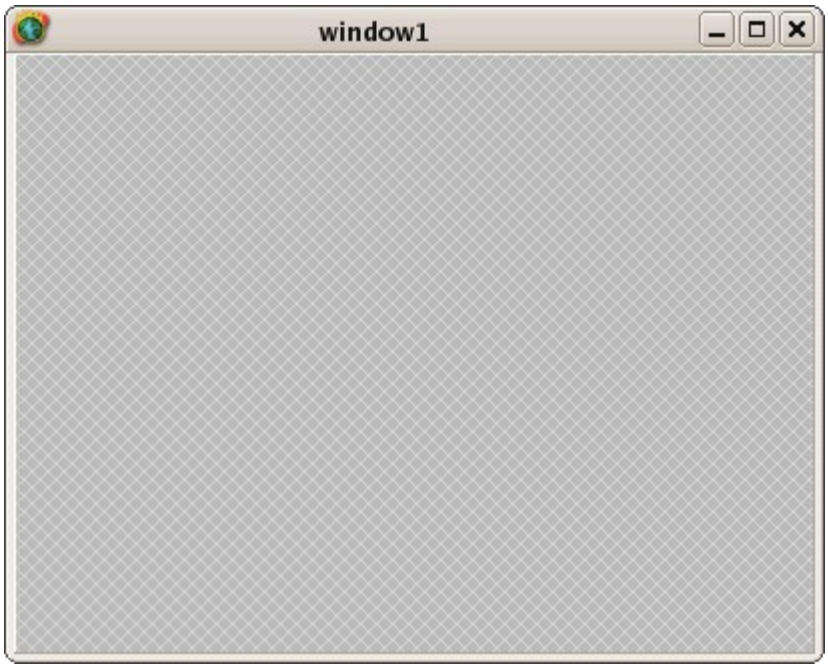


新项目中有 New GTK+ Project 和 New GNOME Project，这二者前者更原始、更基础，适用性也更广，甚至在 MSWindows 系统中也有它的开发系统 ；后者也许功能更丰富，但需要你在 Linux 下运行 GNOME 桌面，本文介绍前者。

选择中间项后就可以开始“涂鸦”了——不要担心，大胆尝试吧！

GTK+内的“涂鸦”有个容器的概念，比如最底层的容器是窗口，窗口里可布设竖向排列的容器，竖立向排列的容器里可布设按钮等部件或再布设横向排列的容器……

这里先需要布设一个窗口容器：点“调色板”窗口里的，来建立一个窗口：

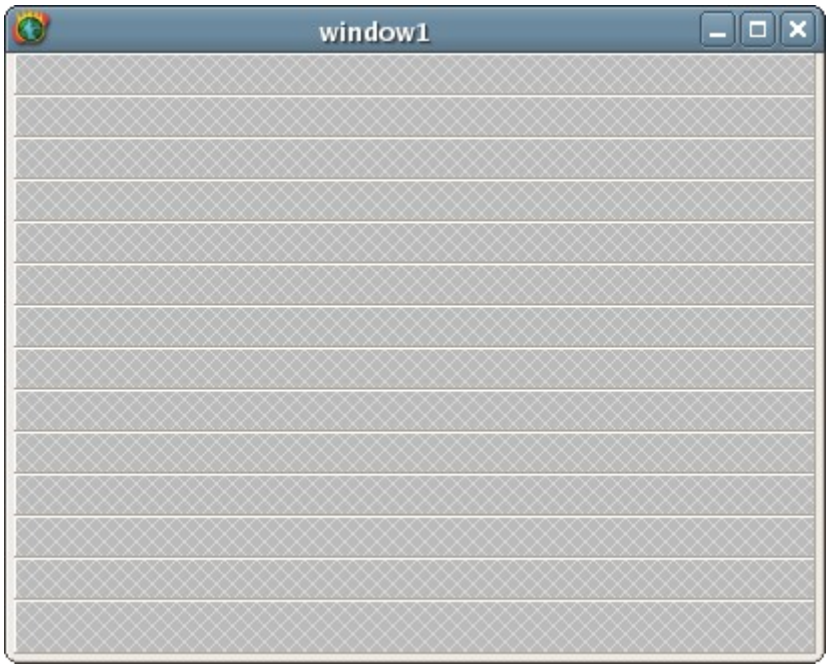


本项目内，我们要有 14 行的内容，所以接着选取竖向容器[垂直框]：

点，接着在上面的 window1 图框内点一下，出现：



将行数 3 改为 14，按[确定(O)]，界面成为：



14 行的竖向容器（或称垂直框）生成了。

在我们的项目中，这 14 行中准备了 1 行的说明，11 行复选按钮，1 行的排列格式（可不用，仅为美观起见），1 行的[Ok][Cancel]按钮，所以，在下面依次操作：

（1）点调色板上的，再点第 1 行，点再点第 1 行 ；

（2）点调色板上的，再点第 2 行 ；

（3）重复（2）步，点 3~12 行 ；

（4）点调色板上的，再点第 13 行 ；

（5）点调色板上的，再点第 14 行，



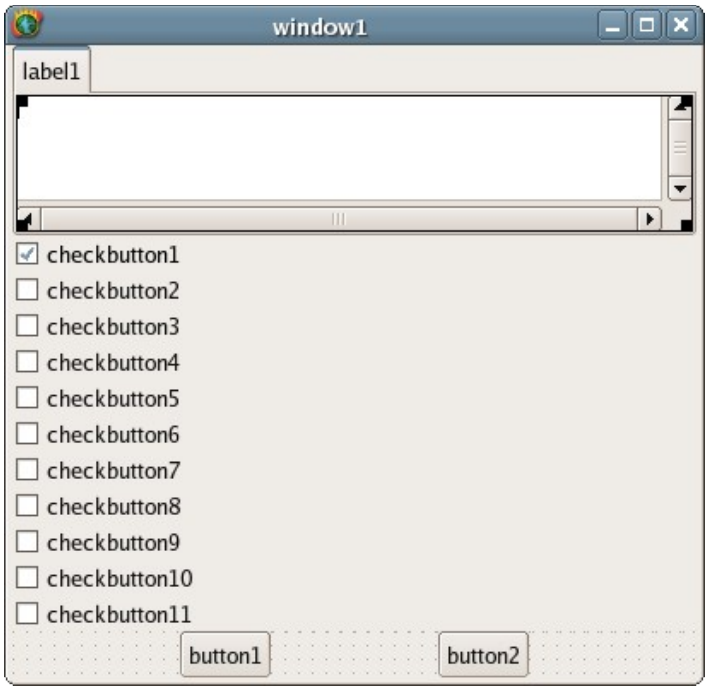
将 3 改为 5 ；

（6）点调色板上的，再点第 14 行的 1、3、5 列 ；

（7）点调色板上的，再点第 14 行的 2、4 列 ；

上面操作后成为：

初步成果可以先保存一下，

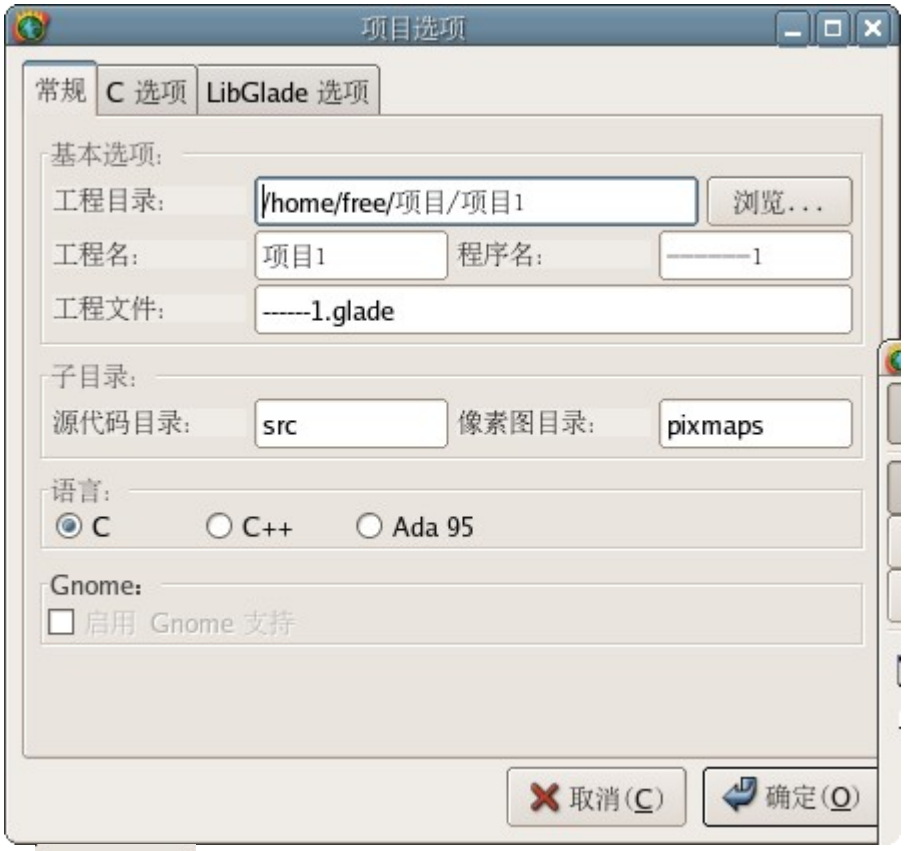


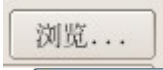
三、保存

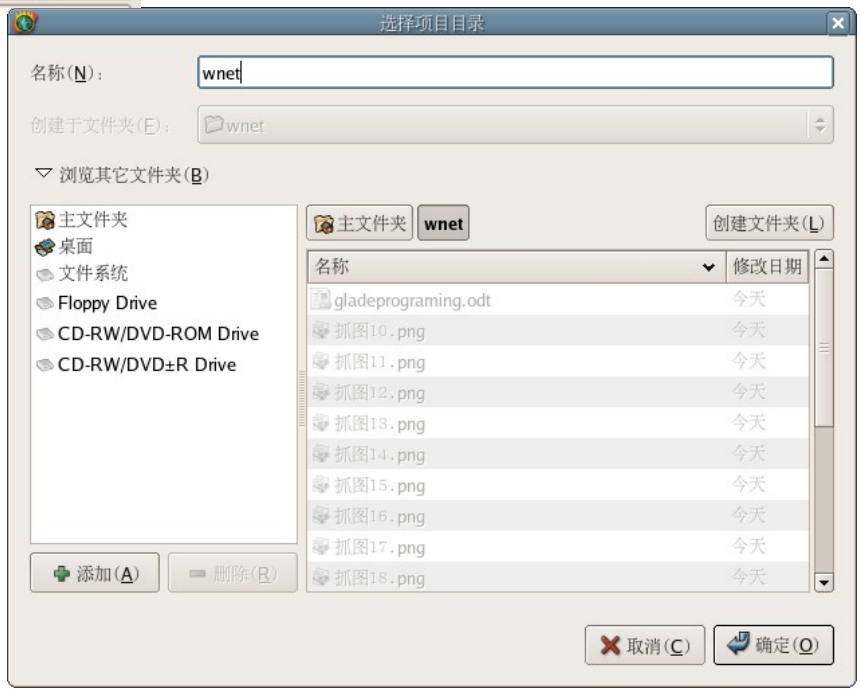
点 

将项目进行保存。

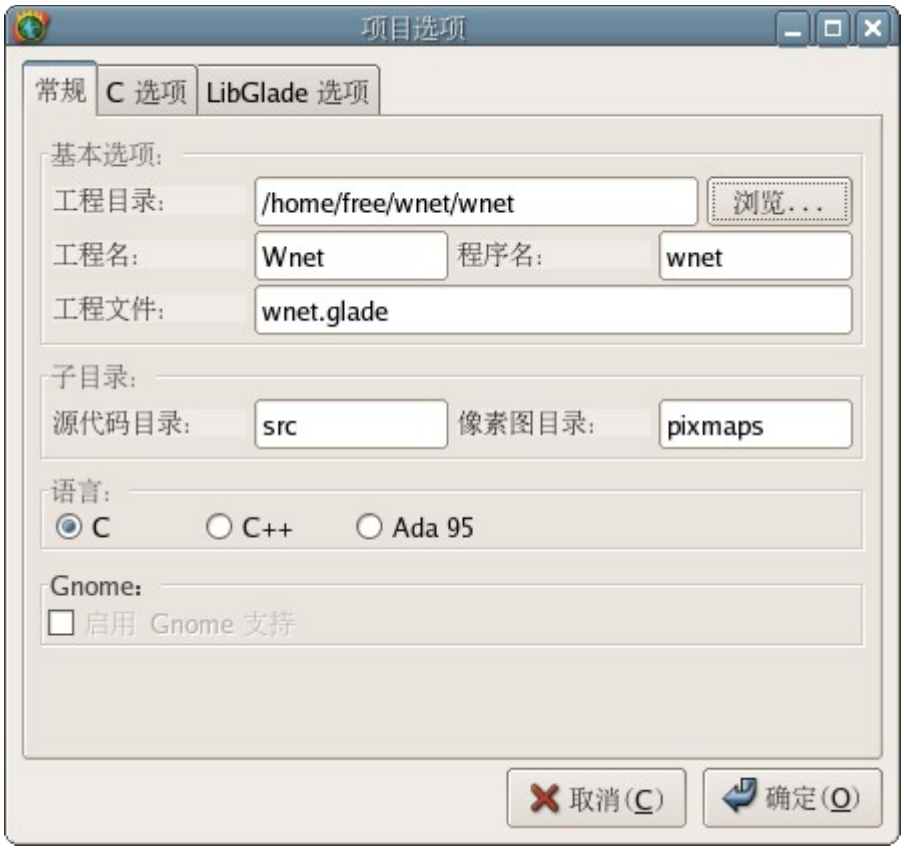
我们应先用[应用程序]->系统工具->文件浏览器创建了一个文件夹，名为 wnet，在出现



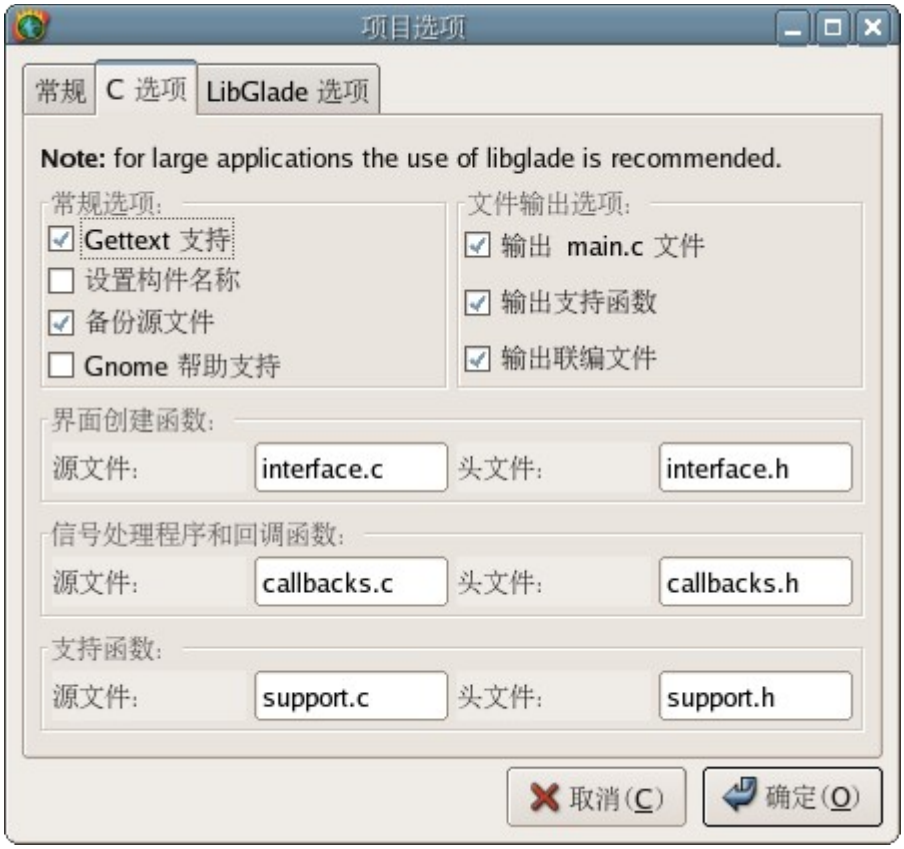
时，点 

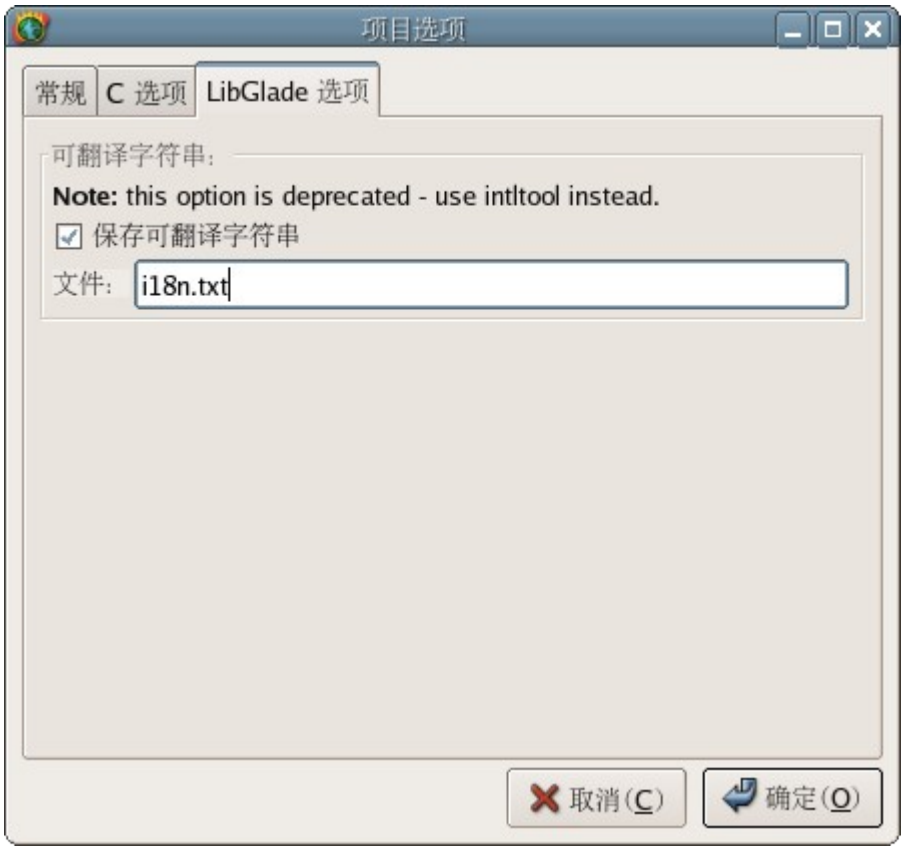


成为



另外 2 个 TAB 页为





四、修改

修改一些参数来适应个性化需要:

1）调出属性对话框

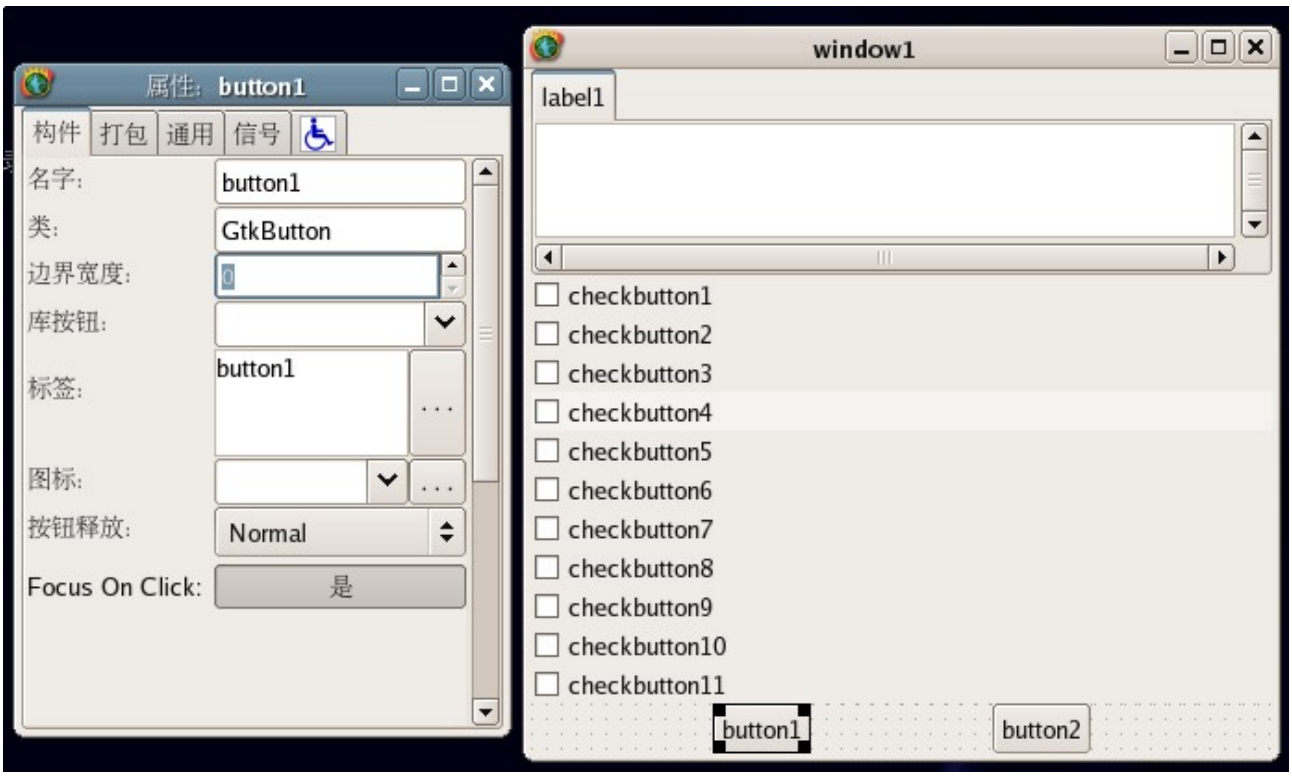
点击 glade 菜单的[查看]->显示属性编辑器

就可以调出属性对话框。

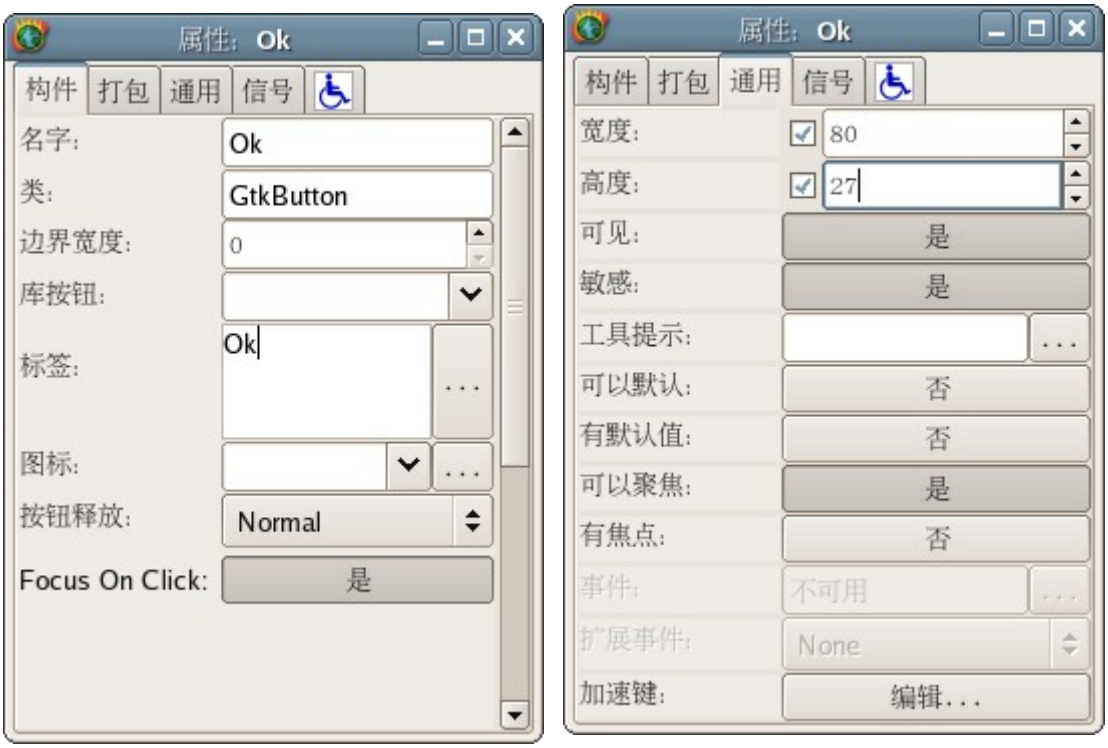
2）修改外观属性

修改通过属性对话框进行 ；

比如调整[button1]改为[Ok]键

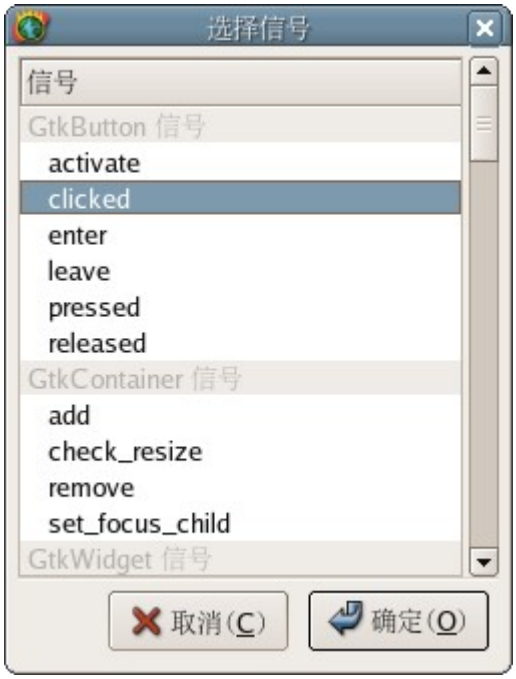


将[构件]栏内的[名字]和[标签]中的 button1 都改为 Ok 即可，宽度和高度可在[通用]栏内设定，这些是外观的修改。



3）修改信号属性

除了修改外观，还需要设定一些信号，在[信号]栏选…，



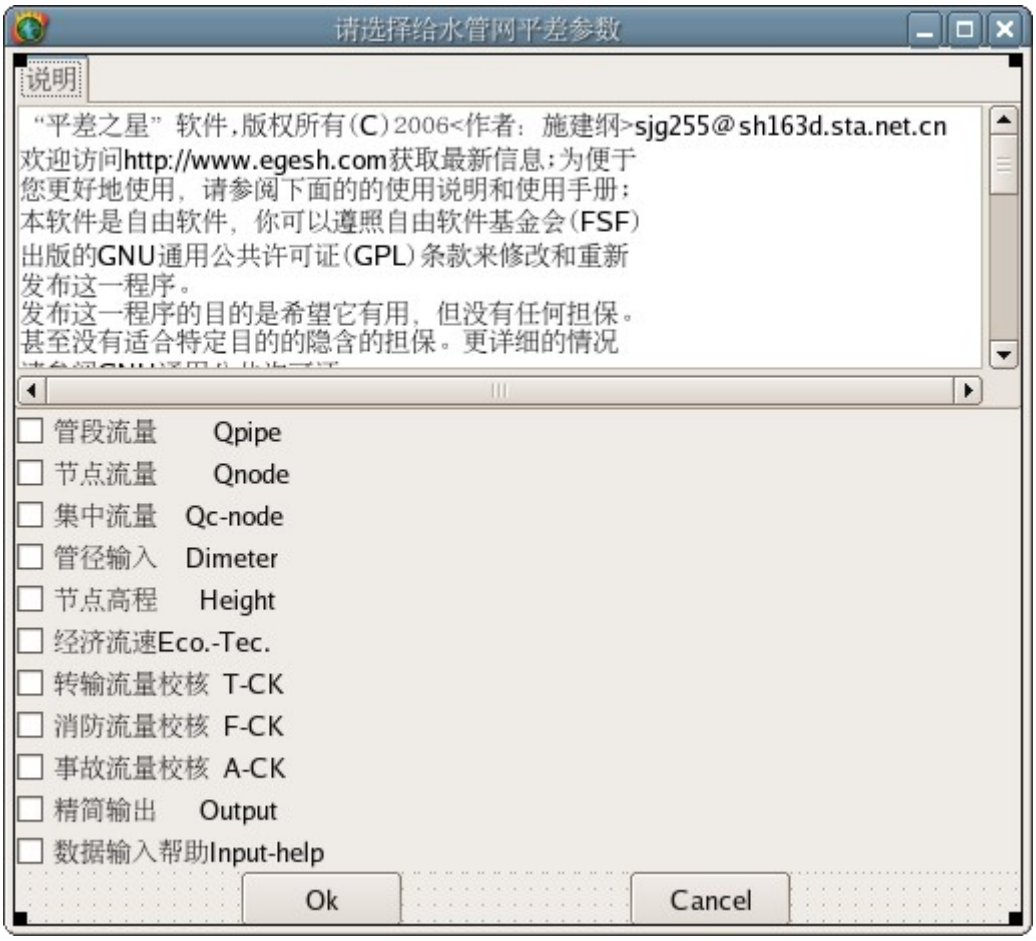


出现此框后再点中“clicked”或其它信号后，按 

，再点[添加]键，



出现如上述的对话框即可，其它各项属性雷同。修改后成为



属性修改后，再次进行保存，然后就可以进行代码的联编生成了。



五、C 代码联编和生成

C 代码的联编很简单，按上图中的 

就可以了。

联编是生成界面的代码，需要通过生成来使各部分成为有机统一体。

本工程的路径是~/wnet，~表示主目录

（1）启动终端

点[应用程序]->系统工具->终端

（2）进入该目录，输入下面命令：

cd ~/wnet

（3）输入下面命令生成界面代码

./autogen.sh

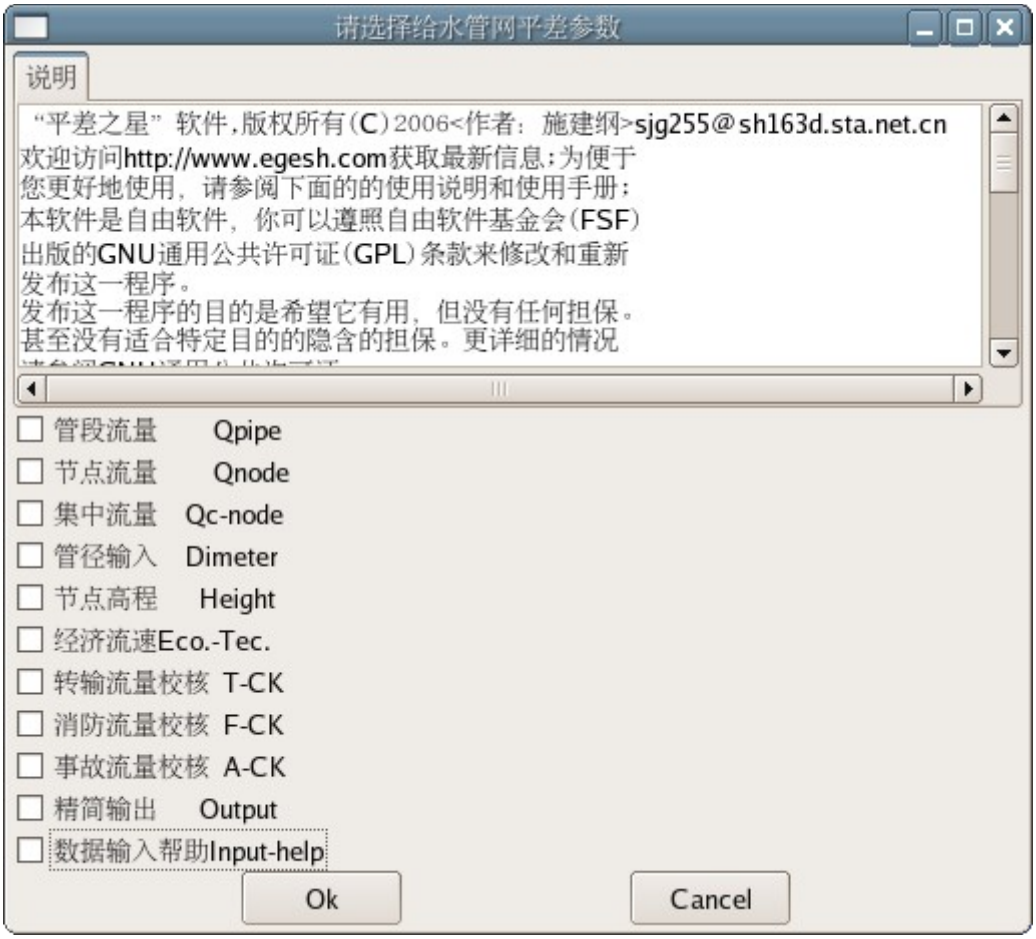
（4）输入下面命令进行编译

make

（5）进入源代码目录运行

cd ~/wnet/src

./wnet



经过上面一系列的体力劳动，图形界面已经生成，运行后就可以看到上面

的界面，已经是一个独立的程序，只是这还是个空架子，缺乏具体内容。

至此也可以看到，这以前的工作还没有所谓编程的成份，对于一个丝毫不懂计算机 C 语言的人，只要照打上面的几个命令和操作即可完成！当然，空壳五界面有什么用？真正建立程序还实用的内容，这部分的工作就需要修改一些反馈函数来实现，就必须要有 C 语言的基础了。

六、修改 callbacks.c 源代码来适应需要的功能

glade 生成的 C 源代码有：main.c，callbacks.c , interface.c , support.c ,还有几个头文件：callbacks.h , interface.h , support.h，原代码位于子目录 src 下

我们只需要修改 callbacks.c 即可完成想要具备的功能。

1）启动文件浏览器

点[应用程序]->系统工具->文件浏览器

2）找到源文件 callbacks.c 双击打开（在 Fedora core 4 下，双击会自动用 gedit 程序打开）

3）修改几个文件

（1）包含应用程序用到的头文件

```
#include<stdlib.h>
#include "stdio.h"
#include "math.h"
#include "string.h"
```

（2）定义全局变量

```
#define Ng 12
int ps[Ng] ;
```

（3）修改选中各复选按钮时的回调函数

```
void
Qpipe_toggled                    (GtkToggleButton *togglebutton,
                                                    gpointer         user_data)
{
if(ps[0]!=1){ps[0]=1 ; } //增加内容
else{ps[0]=0 ; } //增加内容，通过修改全局变量来传递参数
g_printf("（1）Qpipe—管段流量输入判断：ps[0]=%d\n",ps[0]) ;
  //通过打印变量回馈值到虚拟终端可方便程序调试
}
```

复选按钮的其余函数雷同。

（4）修改退出函数

```
void
on_Cancel_clicked                (GtkButton       *button,
                                  gpointer         user_data)
{
gtk_main_quit() ;    //这是一句万能膏药，帖在“正常取消退出”的子程序里
}
```

（5）完成功能函数的开发

```
void
on_Ok_clicked                    (GtkButton       *button,
                                  gpointer         user_data)
{
//在这里放置功能程序的全部内容，详细内容略
}
```

4）重编译

（1）进入工作目录，输入下面命令：

```
cd ~/wnet
```

（2）输入下面命令生成界面代码

```
./autogen.sh
```

（3）输入下面命令进行编译

```
make
```

（4）进入源代码目录运行

```
cd ~/wnet/src
./wnet
```

此时，界面与前面没有两样，但是程序可以实际发挥作用了，已经具备了具体内容。

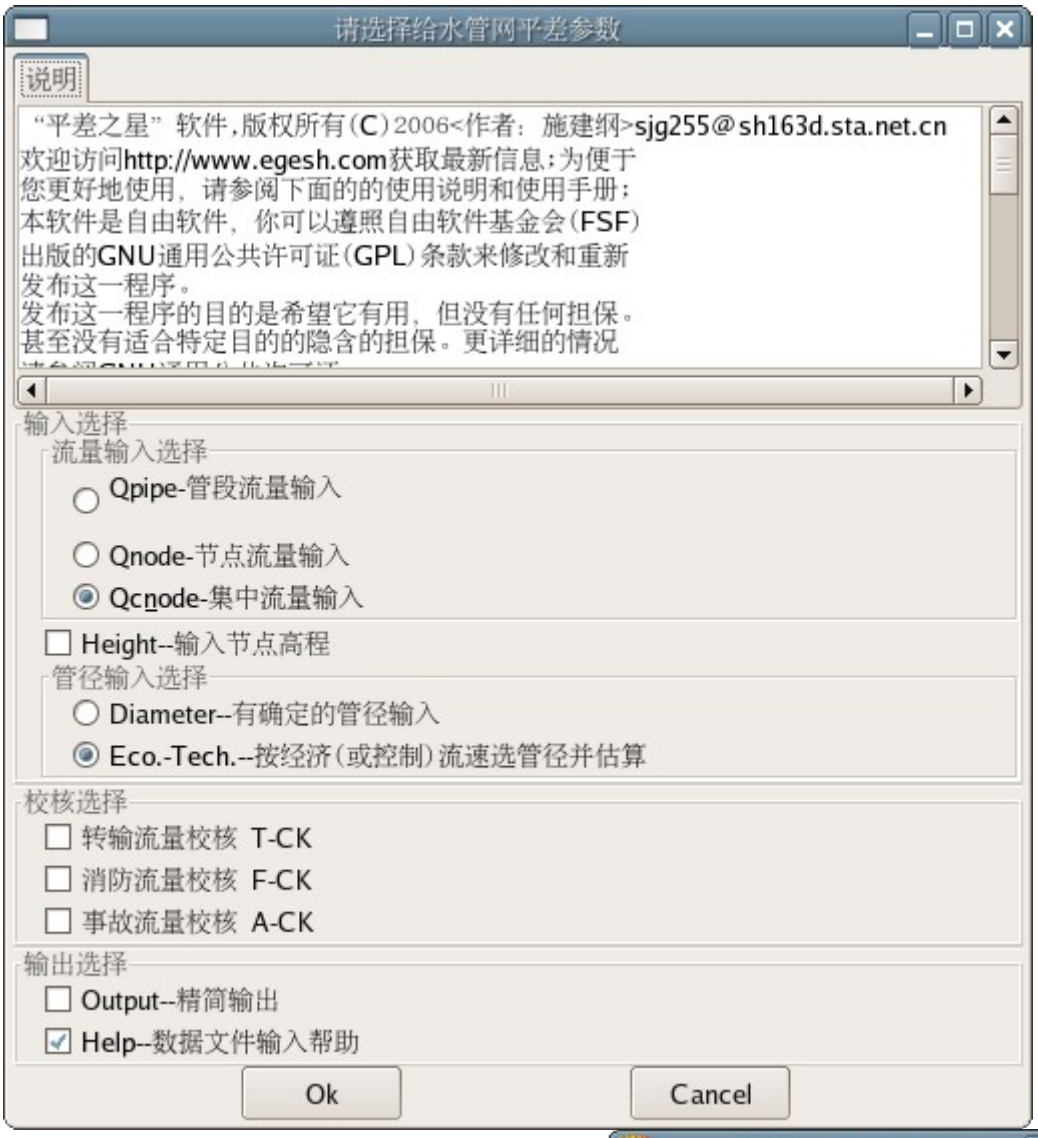
七、美化

上面已完成了基本的图形界面编程的步骤，可以看到，整个过程出乎意料的简单，只用了一千左右的字数来进行描述，相信一个新手经过上面的操作一定信心倍增！

当然这个界面还是相对粗糙的，有兴趣的朋友可以继续进行美化的工作，美化还是通过属性对话框来实现。通过用[单选按钮]替换[复选按钮]，增加一些

框架和分隔线，就可以让这个程序的层次更加分明。

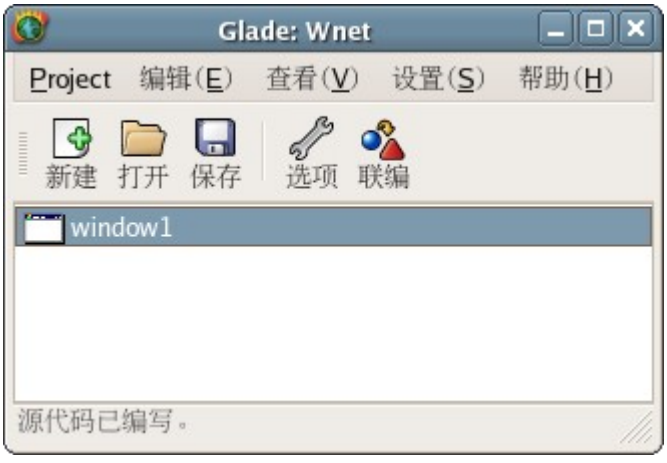
下面是调整运行后的结果



下面是调整的一些中间步骤，列举以作参考：

1）启动 glade ；

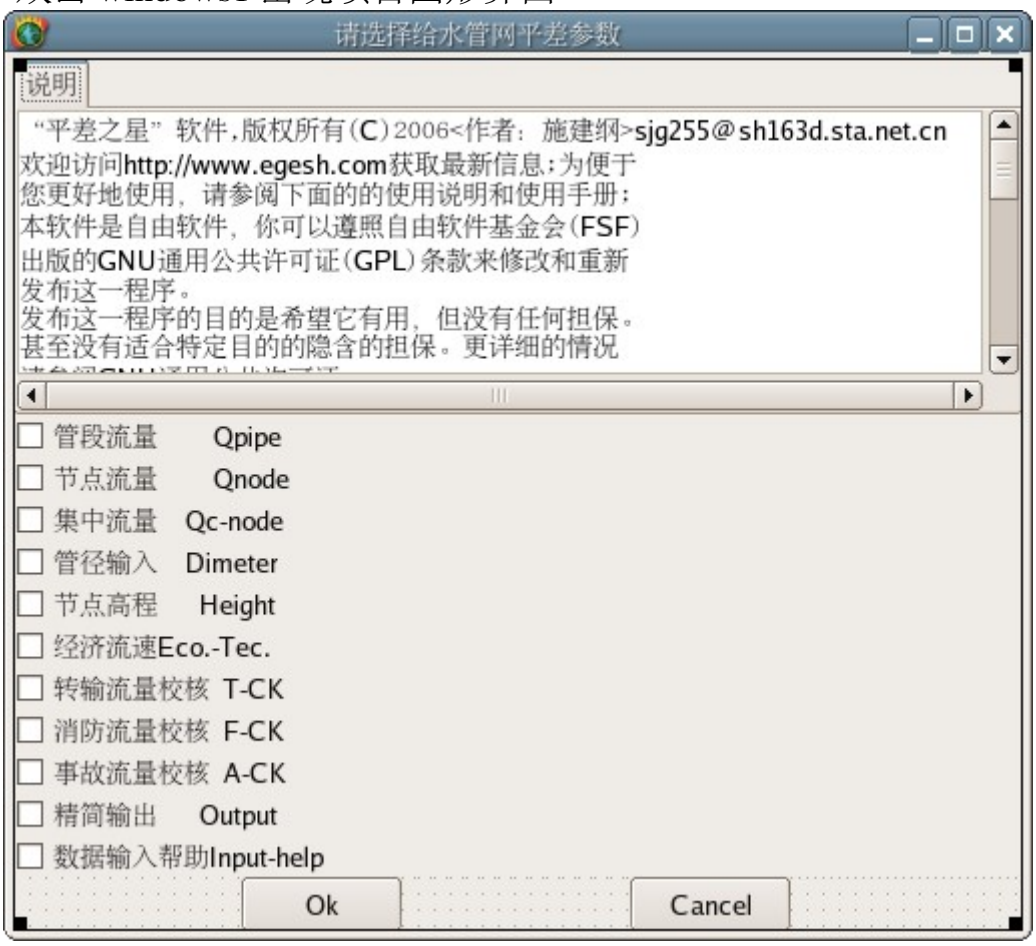
2）点菜单上的[Project]->[打开]来打开 wnet 项目 ；



3）点[查看]->[显示属性编辑器]



4）双击 windows1 出现项目图形界面



5）左键点中[管段流量]行，右键后出现弹出菜单

点[vbox1]->[在以下对象之前插入]；

6）点 glade 调色板上的 帧图标 ；

7）点 5）步生成的容器

8）点 glade 调色板上的 再点中 7）步中的容器，行数取默认的 3 行 ；

9） 点 glade 调色板上的 再点中 8）生成的 1 行容器 ；

10)按 9）步操作两次生成 3 行单选按钮 ；

11）然后点中 6）、7）、8）、9）、10）步中生成的各元素进行属性修改，修改后的属性如下面的图示：





frame2 中的[Etched In]显示一条凹形的框线（你也可以选择突出的框线）；label3 的属性中可以输入框中的内容。



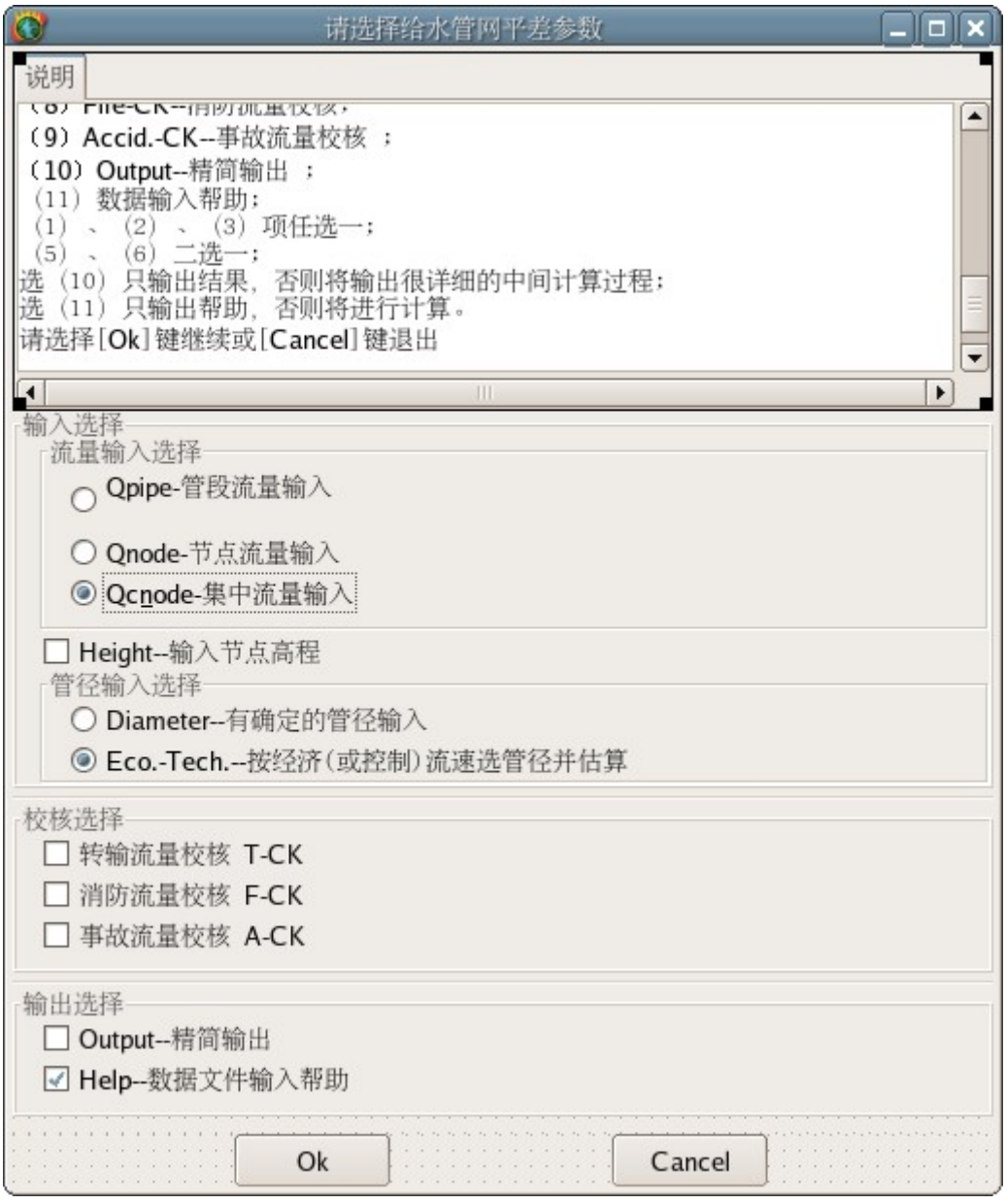


单选按钮 radiobutton1 和 radiobutton2 是由原来的复选按钮修改过来的，其差别在于复选按钮本身独立，而单选按钮是按组相关的，每组中只能有一个初始时打开，注意同组单选按钮[编组]一行的内容须相同，如果须新生成一组时

须选择[New Group]项。

技巧：在项目图形界面的相应“容器”上右键弹出菜单，然后选择相应的操作是修改图形界面和属性的基本方法，你可以增加、删除、复制，如要移动，可以先剪切，再粘贴来实现，仔细体味和操作你会发现这方面 glade 相当强大实用。

12）其余的界面和属性修改与 5）~11）步雷同，不赘述了，修改后界面如下：



至此，图形界面已经编辑完成，如果在修改属性时注意保持信号反馈函数名与本文第六部分保持统一的话，对 callbacks.c 源代码可以保持而不必更改。

保持 callbacks.c 源代码可以使用两种方法：

第一种方法是用文件浏览器找到 callbacks.c，再双击打开它，然后在本文前面第六部分所述的[保存]、[联编]、生成代码后，在保存一下打开的

callbacks.c，然后再进行 make 操作。

第二种方法是在本文前面第六部分所述的[保存]、[联编]、make，按出错信息将后生成的一些内容删除。

推荐第一种方法。

13）至此，如果没有什么意外，用 glade 对市政给水管网平差图形界面程序的开发就完成了，希望也应该 Glade makes you glad!

九、GTK+的运行机理

尽管 Glade 使基于 GTK+库图形界面的 C 代码生成变得简单方便，了解一些 GTK+程序的一些基本组成还是必须的，否则，那只是在玩，而不是在做事。

一）主程序结构

看看附录中 main.c 程序，程序开始后申明一个结构指针

GtkWidget *window1;//这是一个 GTK+的专用结构，所有的如窗口、按钮、标签等都需要这样申明

后面

gtk_init (&argc, &argv);//初始化 GTK，使你在终端窗口命令行参数传递给主程序

window1 = create_window1 ();//由子程序生成界面

gtk_widget_show (window1);//显示界面

gtk_main ();//主程序进入循环并监听信号，这一句十分重要，没有这一句则程序运行时会一闪而过

return 0;//程序返回 0 并结束

二）信号与回调函数

GTK+以信号作为主程序与子程序交互、人机交互的通信手段，信号的种类在前面属性修改对话框中可以一览无余，如点击、按下、放开、离开等等。由不同信号联结 callbacks 中的不同函数，具体语句可在 interface.c 中见到，如对于按下[取消]按钮的信号联结语句

```
g_signal_connect ((gpointer) Cancel, "clicked",
                  G_CALLBACK (on_Cancel_clicked),
                  NULL);
```

在这里第一个参数(gpointer) Cancel 是前面已定义的 GTK+按钮指针

第二个参数"clicked"是信号

第三个参数 on_Cancel_clicked 是回调函数，在 callbacks.c 中定义

第四个参数 NULL 是传给回调函数的参数，它是一个指针，对应于回调波

函数中的 gpointer user_data，如在 callbacks.c 中 on_Cancel_clicked 波函数代码如下：

```
void
on_Cancel_clicked              (GtkButton       *button,
                                gpointer         user_data)
{
gtk_main_quit();

}
```

对应于第二个参量

三）事件

事件是窗口管理器而非应用程序的信号，它是一个枚举类型，共有 16 种：

GDK_DELETE 窗口删除事件

GDK_DESTROY 窗口完全取消并从内存中删除

GDK_EXPOSE 显示窗口

GDK_ON_EXPOSE 需要刷新窗口

GDK_VISIBILITY_NOTIFY 窗口可视状态变化，如隐藏变显示

GDK_MOTION_NOTIFY 鼠标移动

GDK_BUTTON_PRESS 鼠标键被按下

GDK_BUTTON_REALEASE 鼠标在窗口内被释放

GDK_KEY_PRESS 在窗口拥有焦点时按下一个键

GDK_KEY_RELEASE 在窗口拥有焦点时释放一个键

GDK_ENTER_NOTIFY 鼠标进入窗口，或窗口变为活动

GDK_LEAVE_NOTIFY 鼠标离开窗口，或窗口变为不活动

GDK_FOCUS_CHANGE 改变当前接收键盘事件的窗口

GDK_RESIZE 重新定义窗口尺寸

GDK_MAP 窗口被映射（使其可视）

GDK_UNMAP 窗口从视图中删除

九、附录（原代码）

下面给出了市政给水管网平差图形界面程序的完整原代码，供参考，代码中自动生成的部分以黑色表示，修改的部分以红色表示，可以看到修改的代码主要为 callbacks.c，main.c 中增加了部分赋值语句，其余均为自动生成的代码。

如果没有 glade，通过这些原代码也可以在 GTK+2.0 中进行编译，不同的是用人脑子干机器的活也许会感觉很累。

```
//callbacks.c
//=====================================================
 #include<stdlib.h>
 #include "stdio.h"
 #include "math.h"
 #include "string.h"
 #define Ng 12
#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <gtk/gtk.h>

#include "callbacks.h"
#include "interface.h"
#include "support.h"
int ps[Ng],i;

void
Qpipe_toggled                  (GtkToggleButton *togglebutton,
                                                gpointer         user_data)
{
if(ps[0]!=1){ps[0]=1;}
else{ps[0]=0;}
}


void
Qnode_toggled                  (GtkToggleButton *togglebutton,
```

```
                                                    gpointer        user_data)
{
if(ps[1]!=1){ps[1]=1;}
else{ps[1]=0;}
}


void
Qc_node_toggled                 (GtkToggleButton *togglebutton,
                                                    gpointer        user_data)
{
if(ps[2]!=1){ps[2]=1;}
else{ps[2]=0;}

}

void
Dimeter_toggled                 (GtkToggleButton *togglebutton,
                                                    gpointer        user_data)
{
if(ps[3]!=1){ps[3]=1;}
else{ps[3]=0;}

}

void
Height_toggled                  (GtkToggleButton *togglebutton,
                                                    gpointer        user_data)
{
if(ps[4]!=1){ps[4]=1;}
else{ps[4]=0;}

}
```

```
void
Eco__Tec__toggled                (GtkToggleButton *togglebutton,
                                        gpointer         user_data)
{
if(ps[5]!=1){ps[5]=1;}
else{ps[5]=0;}

}


void
T_CK_toggled                (GtkToggleButton *togglebutton,
                                        gpointer         user_data)
{
if(ps[6]!=1){ps[6]=1;}
else{ps[6]=0;}

}


void
F_CK_toggled                (GtkToggleButton *togglebutton,
                                        gpointer         user_data)
{
if(ps[7]!=1){ps[7]=1;}
else{ps[7]=0;}

}


void
A_CK_toggled                (GtkToggleButton *togglebutton,
                                        gpointer         user_data)
{
```

```
if(ps[8]!=1){ps[8]=1;}
else{ps[8]=0;}

}


void
Output_toggled                  (GtkToggleButton *togglebutton,
                                                gpointer         user_data)
{
if(ps[9]!=1){ps[9]=1;}
else{ps[9]=0;}

}


void
Input_help_toggled                (GtkToggleButton *togglebutton,
                                                gpointer         user_data)
{
if(ps[10]!=1){ps[10]=1;}
else{ps[10]=0;}
/*g_printf("（1）Qpipe--管段流量输入判断：ps[0]=%d\n",ps[0]);
g_printf("（2）Qnode--节点流量输入判断：ps[1]=%d\n",ps[1]);
g_printf("（3）Qc_node--集中流量输入判断：ps[2]=%d\n",ps[2]);
g_printf("（4）Diameter--是否有确定的管径输入判断：ps[3]=%d\n",ps[3]);
g_printf("（5）Height--是否输入节点高程判断：ps[4]=%d\n",ps[4]);
g_printf("（6）Eco.-Tech.--按经济(或控制)流速选管径并估算判断：
ps[5]=%d\n",ps[5]);
g_printf("（7）Trans.-CK--转输流量校核判断：ps[6]=%d\n",ps[6]);
g_printf("（8）Fire-CK--消防流量校核判断：ps[7]=%d\n",ps[7]);
g_printf("（9）Accid.-CK--事故流量校核判断：ps[8]=%d\n",ps[8]);
g_printf("（10）Output--精简输出判断：ps[9]=%d\n",ps[9]);
g_printf("（11）Help—数据文件输入帮助判断：ps[10]=%d\n\n",ps[10]);*/
  //通过打印变量回馈值到虚拟终端可方便程序调试
```

```
}



void
on_Ok_clicked                  (GtkButton       *button,
                                gpointer         user_data)
{
 double pow();
 double fabs();
 typedef struct src *qsrc1;
 typedef struct Hstr *Hb1;
 struct Hstr
 {
 float Habs;
 float Hfre;
 int jus;
 };
 struct src
 {
 int Psrc;
 int Nsrc;
 float Qsrc;
 float Hsrc;
 };

 qsrc1 qsrc;
 Hb1 Hb;

 char ch;
 int M,M1,M2,M3,N,KO,js,i,j,k,im,sz,ip,iq,id,N1,ir,ns,is,Pbad,Nf,Ks;
 int *IO,*JO,**pipe,*Dj,Nof[6],**pipe1,jjt;
 float *L,*D,*q,*Q,*S,*H,*V,Vk,*q2;
 float *F,*R,*DQ,*DI;
 int Px,Pb,Pz,Pt,Pp=0;
```

```
 float eps,K3,PI=3.141593,A1,V1,A2,V2,C,costb,Qsum,Lsum,Lp,qs,Hs,Xz;
/*       static float DI1[19]={1.5,1.4,1.3,1.2,1.1,1.,.9,.8,.7,.6,.5,.45,.4,.35,
 .3,.25,.2,.15,.1};*/
 static float DI2[10]={1.2,1.,.9,.8,.7,.6,.5,.4,.3,.2};


 float gama,E,p,t,a,b,n,st1,st2,st3,st4,is1;
 float *DI3,*Cp,*QI1;
 float   m=5.33,kk=1.48,f,costb1,Wf;
 float depth1,depth2,depth3,wide,soil,fill,base,pad;
 int  iu;
 void box1();
 int g;
 union k{char c[2];int j;}key;
 int X1=35;
 int locy[Ng];
 Pb=0;
 Pz=0;
if(ps[9]==0){
/*printf("image cicles(N1),pipes(M),cicles(N),diameter ajust loop(M2):\n");*/
printf("\n 请输入水源数(N1),管数(M),环数(N),节点数(M1),调整管径的遍数
(M2):(整数)\n");
    }
 scanf("%d%d%d%d%d",&N1,&M,&N,&M1,&M2);
 printf("%d 水源  %d 管  %d 环  %d 节点  调整管径%d 遍
\n",N1,M,N,M1,M2);
if(ps[9]==0){
/*printf("image cicles(N1),pipes(M),cicles(N),diameter ajust loop(M2):\n");*/

/*printf("Input the %d water source pipe No.,Node
No.,Quantity,Presure:\n",N1);*/
 printf("请输入这  %d 个水源管号,节点号,流量,水压:(整,整,实,实)\n",N1);
/*       printf("Input pipe length-->L[%d](m):\n",M);*/
 printf("请输入各管长(实数)-->L[%d](m):\n",M);
/*       printf("Input inside cicle No.-->IO[%d]\n",M);*/
```

```
        printf("请输入各内环号(整数)[顺时针流向为正,逆为负,不许为 0]-->IO[%d]\n",M);
        /*      printf("Input outside cicle No.-->JO[%d]\n",M);*/
         printf("请输入各外环号(整数)-->JO[%d]\n",M);
        /*printf("Input the node No. of pipe & character value(0/1)-->
         pipe[%d][3]\n",M);*/
        printf("请输入各管段水流的起、止节点号及供水特征值（0/1/2）(整数)-->pipe[%d][3]\n",M);
        printf("    [管段水流的起止表示水流方向;管段号同输入节点的顺序;\n");
        printf("    供水特征值双侧为 0,单侧为 1,不供为 2]\n");
        if(ps[4]==1)printf("请输入该 %d 节点标高\n",M1);
         if(ps[0]==1)
        {
        /*      printf("Input the pipe quantity-->Q[%d](m3/s)\n",M);*/
         printf("请输入各管段流量(实数)-->Q[%d](m3/s)\n",M);
        }
         if(ps[0]==0)
        {
         if(ps[1]==1||ps[2]==1)
          {
        /*      printf("Input the node quantity-->q[%d](m3/s)\n",M);*/
         printf("请输入各节点流量(实数)-->q[%d](m3/s)\n",M1);
          }
        }
         if(ps[3]==1)
          {
        /*      printf("Input the pipe diameter-->D[%d](m)\n",M);*/
         printf("请输入各管段管径(实数)[未定管径取 0.0]-->D[%d](m)\n",M);
          }
        /*      printf("Input the value of Dg400 pipe-->costb(Yuan/m)\n");*/
         printf("请输入 Dg400 管的单位长度价格(实数)-->costb(元/m)\n");
        /*      printf("Input control virosity Vk(m/s):\n");*/
         printf("请输入确定管径的控制流速 Vk(m/s)及闭合差 eps[0.01](mH2O):\n");
        printf("请输入覆土深度 depth1[0.7](m),基础平均厚 depth2[0.3](m),\n");
```

```
        printf("    垫层厚 depth3[0.3](m),及施工挖槽富余半宽 wide[0.6](m):(实数)
\n");
         if(ps[5]==1)
        {

        /*      printf("Input parameter (gama)[If there is a water tower before\n");
        printf("       net ,the value will be 0.5-0.7;others 0.1-0.4]:\n");*/
        printf("请输入供水能量系数 gama(实数)[网前水塔为 0.5-0.75,其余 0.1-
0.4]:\n");
         /*printf("Input electricity cost E(fen/kw/hr)& pon station
              effecience n(.55-.85):\n");*/
         printf("请输入电费 E(分/度)和泵站效率 n[0.55-0.85]:(实数)\n");
        /*printf("Input investment period t(year);consumer rate[2.8-4.0](%):\n");*/
         printf("请输入投资偿还期 t(年)和管网年折旧率[2.8-4.0](%):(实数)\n");
        /*      printf("Input pipe diameters DI3[i](m)(series from small \n");
         printf("           to large,end with 0.0):\n");*/
         printf("请输入管径系列 DI3[i](m)(实数)[由小到大,以 0.0 结束]:\n");
        /*      printf("Input the cost of these pipes Cp[i](yuan/m):\n");*/
         printf("请输入相应系列管子的单价 Cp[i](元/m)(实数):\n");

        }
         if(ps[6]==1)
        {
         printf("请输入水塔节点号 Pt(整数),转输系数 Xz[Qz/Qmax](实数)-->\n");
        }
         if(ps[7]==1||ps[8]==1)
        {
         printf("请输入消防时的同时失火点数 Nf(整数)及灭火水量 Wf(实数):\n");
         printf("请输入火灾节点 Nof[i](整数):\n");
        }
         if(ps[8]==1)
        {
        /*      printf("Input the damaged pipe No.-->Pbad\n");*/
         printf("请输入事故管段的管段号-->Pbad\n");
        }
```

```
    }
 if(ps[10]==1)exit(0);
 sz=sizeof(struct src)*N1;
 qsrc=(qsrc1)malloc(sz);

 if(!(qsrc))
{
 printf("qsrc 内存不够!");
 exit(1);
}
 sz=sizeof(struct Hstr)*M1;
 Hb=(Hb1)malloc(sz);
 if(!(Hb))
  {
 printf("Hb 内存不够!");
 exit(1);
  }
 for(i=0;i<N1;i++)
 {
/*printf("Input the %d water source pipe No.,Node
No.,Quantity,Presure:\n",N1);
 printf("请输入 %d 个水源管号,节点号,流量,水压:\n",N1);*/
scanf("%d%d%f%f",&qsrc[i].Psrc,&qsrc[i].Nsrc,&qsrc[i].Qsrc,&qsrc[i].Hsrc);
if(ps[9]==0)printf("%d %d %f %f\n",
 qsrc[i].Psrc,qsrc[i].Nsrc,qsrc[i].Qsrc,qsrc[i].Hsrc);
 }
/*      M1=M-N+1;*/
 sz=M*sizeof(float);
 L=(float *)malloc(sz);
 D=(float *)malloc(sz);
 Q=(float *)malloc(sz);
 S=(float *)malloc(sz);
 H=(float *)malloc(sz);
 V=(float *)malloc(sz);
 sz=M1*sizeof(float);
```

```
 q=(float *)malloc(sz);
/*       q1=(float *)malloc(sz);*/
 q2=(float *)malloc(sz);
/*       pt=(float *)malloc(sz);*/
 sz=N*sizeof(float);
 F=(float *)malloc(sz);
 R=(float *)malloc(sz);
 DQ=(float *)malloc(sz);
 sz=M*sizeof(int);
 Dj=(int *)malloc(sz);
 IO=(int *)malloc(sz);
 JO=(int *)malloc(sz);
 sz=sz*3;
 pipe=(int **)malloc(sz);
 sz=M*sizeof(int)*2;
 pipe1=(int **)malloc(sz);
 if(!(L&&Q&&S&&H&&F&&R&&DQ&&IO&&JO&&pipe))
{
printf("Not enough memory!\n");
exit(1);
}
/*       printf("Input pipe length-->L[%d](m):\n",M);
 printf("请输入各管长(实数)-->L[%d](m):\n",M);*/
 for(i=0;i<M;i++)
 scanf("%f",L+i);
if(ps[9]==0) printf("L[%d]%f\n",M,L[M-1]);

/*       printf("Input inside cicle No.-->IO[%d]\n",M);
 printf("请输入各内环号(整数)-->IO[%d]\n",M);*/
 for(i=0;i<M;i++)
 scanf("%d",IO+i);
if(ps[9]==0) printf("IO[%d]%d\n",M,IO[M-1]);

/*       printf("Input outside cicle No.-->JO[%d]\n",M);
 printf("请输入各外环号(整数)-->JO[%d]\n",M);*/
```

```
 for(i=0;i<M;i++)
 scanf("%d",JO+i);
if(ps[9]==0) printf("JO[%d]%d\n",M,JO[M-1]);

/*printf("Input the node No. of pipe & character value(0/1)-->
 pipe[%d][3]\n",M);
printf("请输入各管段的起、止节点号及特征值（0/1）(整数)--
>pipe[%d][3]\n",M);*/
 for(i=0;i<3;i++)
{
 for(j=0;j<M;j++)
  {
 scanf("%d",&pipe[i][j]);
  }
}
if(ps[9]==0) printf("M=%d pipe[2][%d] %d\n",M,M,pipe[2][M-1]);

 for(i=0;i<M;i++)
{
 D[i]=0.0;
 q[i]=0.0;
}
/*printf("请输入该 %d 节点标高\n",M1);
if(ps[4]==1)*/for(i=0;i<M1;i++)
{
 Hb[i].Habs=0.0;
 Hb[i].Hfre=0.0;
 Hb[i].jus=0;
}
if(ps[4]==1)for(i=0;i<M1;i++)
{
 scanf("%f",&Hb[i].Habs);
 if(ps[9]==0){printf("%4.1f",Hb[i].Habs);
 if(i%5==4)printf("\n");}
}
```

```
 if(ps[0]==1)
{
/*       printf("Input the pipe quantity-->Q[%d](m3/s)\n",M);
 printf("请输入各管段流量(实数)-->Q[%d](m3/s)\n",M);*/
 for(i=0;i<M;i++)
 scanf("%f",Q+i);
if(ps[9]==0)  printf("Q[%d]%f\n",M,Q[M-1]);
}

 if(ps[0]==0)
{
 if(ps[1]==1||ps[2]==1)
  {
/*       printf("Input the node quantity-->q[%d](m3/s)\n",M);
 printf("请输入各节点流量(实数)-->q[%d](m3/s)\n",M);*/
 for(i=0;i<M1;i++)
 scanf("%f",q+i);
if(ps[9]==0)  printf("q[%d]%f\n",M1,q[M1-1]);
  }
}

 if(ps[3]==1)
  {
/*       printf("Input the pipe diameter-->D[%d](m)\n",M);
 printf("请输入各管段管径(实数)-->D[%d](m)\n",M);*/
 for(i=0;i<M;i++)
 scanf("%f",D+i);
if(ps[9]==0)  printf("D[%d]%f\n",M,D[M-1]);
  }


 for(i=0;i<M;i++)
  {
 if(D[i]<1e-2)
   {
```

```
        Pp=1;
        Dj[i]=0;
          }
        else Dj[i]=1;
         }

       /*       printf("Input the value of Dg400 pipe-->costb(Yuan/m)\n");
        printf("请输入 Dg400 管的单位长度价格(实数)-->costb(元/m)\n");*/
        scanf("%f",&costb);
       if(ps[9]==0)  printf("costb=%f\n",costb);
       /*       printf("Input control virosity Vk(m/s):\n");
        printf("请输入确定管径的控制流速 Vk(m/s)及闭合差 eps(mH2O):\n");*/
        scanf("%f%f",&Vk,&eps);
        if(ps[9]==0)printf("Vk=%f , eps=%f\n",Vk,eps);
       /*printf("请输入覆土深度 depth1[0.7](m),基础平均厚 depth2[0.3](m),\n");
       printf("    垫层厚 depth3[0.3](m),及施工挖槽富余半宽 wide[0.6](m):(实数)
\n");*/
        scanf("%f%f%f%f",&depth1,&depth2,&depth3,&wide);
        if(ps[9]==0)printf("depth1=%f depth2=%f depth3=%f wide=%f\n",
              depth1,depth2,depth3,wide);

        if(ps[5]==1)
       {

       /*       printf("Input parameter (gama)(If there is a water tower before\n");
       printf("       net ,the value will be 0.5-0.7;others 0.1-0.4):\n");
       printf("请输入供水能量系数 gama(实数)[网前水塔为 0.5-0.75,其余 0.1-
0.4]:\n");*/
        scanf("%f",&gama);
        /*printf("Input electricity cost E(fen/kw/hr)& pump station
            effecience n[.55-.85]:\n");
        printf("请输入电费 E(分/度)和泵站效率 n[0.55-0.85]:(实数)\n");*/
        scanf("%f %f",&E,&n);
       /*printf("Input investment period t(year);consumer rate[2.8-4.0](%):\n");
        printf("请输入投资偿还期 t(年)和管网年折旧率 p[2.8-4.0](%):(实数)\n");*/
```

```
 scanf("%f%f",&t,&p);
if(ps[9]==0) printf("gama %f E %f n %f t %f p %f\n",gama,E,n,t,p);
 DI3=(float *)malloc(M3*sizeof(float));
 Cp=(float *)malloc(M3*sizeof(float));
 QI1=(float *)malloc(M3*sizeof(float));
 if(!(DI3&&DI))
  {
 printf("Not enough memory!");
 exit(1);
  }
 costb1=12./83.6*costb;
/*       printf("Input pipe diameters DI3[i](m)(series from small \n");
 printf("           to large,end with 0.0):\n");
 printf("请输入管径系列 DI3[i](m)(实数)[由小到大,以 0.0 结束]:\n");*/
 iu=0;
 do
  {
 scanf("%f",&DI3[iu]);
 iu++;
  }while(DI3[iu-1]>1e-3);
 iu--;
/*       printf("Input the cost of these pipes Cp[i](yuan/m):\n");
 printf("请输入相应系列管子的单价 Cp[i](元/m)(实数):\n");*/

 for(i=0;i<iu;i++)
  {
 scanf("%f",&Cp[i]);
if(ps[9]==0)printf("DI3[%d] %f %f\n",i,DI3[i],Cp[i]);
  }
 costb1=.618*Cp[0];
 for(i=0;i<iu;i++)
  {
 Cp[i]=log(Cp[i]-costb1);
 DI3[i]=log(DI3[i]);
  }
```

```
        st1=0.0;
        st2=0.0;
        st3=0.0;
        st4=0.0;
        for(i=0;i<iu;i++)
         {
        st1=st1+DI3[i];      /*sum logx*/
        st2=st2+Cp[i];       /*sum logy*/
        st3=st3+DI3[i]*DI3[i];/*sum logx**2*/
        st4=st4+DI3[i]*Cp[i];/*sum logxlogy*/
         }
        is1=(float)(iu);
        a=(st2/is1-st4/st1)/(st1/is1-st3/st1);
        b=st2/is1-st1/is1*a;
        b=exp(b);
       if(ps[9]==0) printf("经济计算公式: value=%6.2f+
%8.2f*D(**%5.3f)\n",costb1,b,a);
        f=86.*gama*E*kk*m/((p+100./t)*a*b*n);
       if(ps[9]==0) printf("经济指数  f=%f\n",f);
       /*       f=1.0;*/
        for(i=0;i<M3-1;i++)
         {
       st1=(pow(DI[i],a)-pow(DI[i+1],a))/(pow(DI[i+1],-m)-pow(DI[i],-m));
        QI1[i]=pow(m/f/a,1./3.)*pow(st1,1./3.);
       if(ps[9]==0){         printf("                Q[%f]界限流量\n",QI1[i]);
        printf("管径 D[%f] -<\n",DI[i]);
            }
         }
       }
        if(ps[6]==1)
       {
       /*       printf("请输入水塔节点号 Pt,转输系数 Xz[Qz/Qmax]-->\n");*/
        scanf("%d%f",&Pt,&Xz);
       if(ps[9]==0) printf("Pt %d Xz %5.3f\n",Pt,Xz);
       }
```

```
 if(ps[7]==1||ps[8]==1)
 {
/*       printf("请输入消防时的同时失火点数 Nf(整数)及灭火水量 Wf(实数):
\n");*/
 scanf("%d%f",&Nf,&Wf);
/*       printf("请输入火灾节点 Nof[i](整数):\n");*/
 for(i=0;i<Nf;i++)
 scanf("%d",&Nof[i]);
 if(ps[9]==0)printf("Nf=%d , Wf=%f , Nof[%d]=%d\n",Nf,Wf,Nf,Nof[Nf-1]);
 }
 if(ps[8]==1)
 {
/*       printf("Input the damaged pipe No.-->Pbad\n");
 printf("请输入事故管段的管段号-->Pbad\n");*/
 scanf("%d",&Pbad);
if(ps[9]==0)  printf("Pbad= %d\n",Pbad);
/*printf("请输入规划名称(字符):\n");*/
 }

 Ks=1;
for(i=6;i<9;i++)Ks=Ks+ps[i];
/*       if(ps[7]==1||ps[8]==1)Ks=2;
 if(ps[7]==1&&ps[8]==1)Ks=3;*/
 for(is=0;is<Ks;is++)
 {
 /*计算总流量*/
 if(ps[0]==0)
  {       if(ps[1]==0)
     {
 Qsum=0.0;
 Lsum=0.0;
 for(i=0;i<N1;i++)
       {
           Qsum=Qsum+qsrc[i].Qsrc;
       }
```

```
 for(i=0;i<M1;i++)Qsum=Qsum-q[i];/*计算总管长*/


 for(i=0;i<M;i++)
       {
 Lsum=Lsum+L[i]*(1.0-0.5*(float)pipe[2][i]);
 if(ps[9]==0)printf("L# %f\n",Lsum);
       }
 qs=Qsum/Lsum;
if(ps[9]==0)  printf("Qsum= %f  Lsum= %f qs= %f \n",Qsum,Lsum,qs);
/*        printf("Primary discharge node quantity q[i](m3/s)\n");*/
if(ps[9]==0)  printf("初分配节点流量  q[i](m3/s)\n");
 /*初分配节点流量*/
 for(i=0;i<M1;i++)
       {
 Lp=0.0;
 for(j=0;j<M;j++)
         {
 for(k=0;k<2;k++)
           {
 if(pipe[k][j]==i)
             {
 Lp=Lp+L[j]*(1.0-0.5*(float)pipe[2][j]);
             }
           }
         }
 q[i]=q[i]+0.5*qs*Lp;
if(ps[9]==0){          printf(" *%d* %8.5f ",i,q[i]);
 if(i%4==3)printf("\n");
     }
       }
      }
if(ps[9]==0)for(i=0;i<M1;i++)
     {
```

```
printf("q[%d]=%8.5f",i,q[i]);
 if(i%4==3)printf("\n");
     }
 /*计算管段流量始*/
 for(i=0;i<M1;i++)q[i]=-q[i];/*令流出节点的流量为负*/
 for(i=0;i<N1;i++)
      {
 q[qsrc[i].Nsrc]=q[qsrc[i].Nsrc]+qsrc[i].Qsrc;
if(ps[9]==0)  printf("q[%d]=%f\n",qsrc[i].Nsrc,q[qsrc[i].Nsrc]);
      }


 for(i=0;i<M;i++)
    {
 Q[i]=0.0;
 pipe[2][i]=0;
    }
        for(i=0;i<N1;i++)
 pipe[2][qsrc[i].Psrc]=1;/*水源管流量分配固定*/
 if(Pb==1)pipe[2][Pbad]=1;/*如果事故校核,令该管流量为 0.0)*/
 for(i=0;i<M1;i++)
    {
 ip=0;
 iq=0;
 ir=0;
 for(j=0;j<M;j++)
      {
 for(k=0;k<2;k++)
       {
 if(pipe[k][j]==i&&pipe[2][j]!=0)
         {
 q[i]=q[i]+Q[j];
         }
 if(pipe[k][j]==i)
         {
```

```
 if(pipe[2][j]==0)
 ip++;
          }
         }
        }
if(ip!=0)       for(j=0;j<M;j++)
       {
 for(k=0;k<2;k++)
         {
 if(pipe[k][j]==i&&pipe[2][j]==0)
           {
 Q[j]=q[i]/ip;
 if((Q[j]<0.0&&k==0)||(Q[j]>0.0&&k==1))
            {
 ir=pipe[0][j];
 pipe[0][j]=pipe[1][j];
 pipe[1][j]=ir;
 IO[j]=-IO[j];
 JO[j]=-JO[j];
            }
 pipe[2][j]=1;
          }
         }
        }
      }
 for(i=0;i<N1;i++)
 Q[qsrc[i].Psrc]=qsrc[i].Qsrc;
   }


if(ps[9]==0){
/*       printf("\nvvvvvvvvvvvvvv *** data file *** vvvvvvvvvvvvvvvv\n");*/
 printf("\nvvvvvvvvvvvvvv *** 数据文件  *** vvvvvvvvvvvvvvvv\n");
 printf("%d 水源  %d 管  & %d 环管网:\n",N1+1,M,N);
/*       printf("%d water source %d pipes  & %d cicle net:\n",N1+1,M,N);*/
```

```
 printf(" i      L(m)   Q(m3/s) IO  JO node[1,2]    q(l/s)     \n");
    }
 for(i=0;i<M;i++)
  {
 Q[i]=fabs(Q[i])*IO[i]/fabs(IO[i]);
}

 if(ps[9]==0)for(i=0;i<M;i++)
{
 if(i<M1)printf(" %3d %9.3f %9.6f %2d %2d %3d,%3d   %9.6f\n",
 i,L[i],Q[i],IO[i],JO[i],pipe[0][i],pipe[1][i],q[i]);
 if(i>=M1)printf(" %3d %9.3f %9.6f %2d %2d %3d,%3d \n",
 i,L[i],Q[i],IO[i],JO[i],pipe[0][i],pipe[1][i]);
}
if(is==0&&ps[8]==1)
 printf("\nvvvvvvvvvvvvvvv *** 结   果  *** vvvvvvvvvvvvvvvvv\n");
/*       printf("\nvvvvvvvvvvvvvvv *** result *** vvvvvvvvvvvvvvvvv\n");*/

 KO=0;
 /*设定管径始*/
 for(id=0;id<M2;id++)
{
/*       printf("---第  %d 次调整管径  D[i](mm)---\n",id);*/
/*       printf("---the  %d time ajust diameter D[i](mm)---\n",id);*/
 for(iq=0;iq<M;iq++)
   {
 if(Dj[iq]==0&&is==0)
     {
 if(ps[5]==0)
       {
 for(ip=0;ip<M3-1;ip++)
         {
 DI=&DI2[0];
 A1=PI*DI[ip]*DI[ip]/4.;
 V1=fabs(Q[iq])/A1;
```

```
A2=PI*DI[ip+1]*DI[ip+1]/4.;
V2=fabs(Q[iq])/A2;

if(fabs(V1-Vk)>fabs(V2-Vk))D[iq]=DI[ip+1];
if(D[iq]==DI[ip])break;
if(ip==(M3-2))D[iq]=DI[M3-2];
        }
      }
if(ps[5]==1)
    {
for(ip=0;ip<M3-1;ip++)
      {
if(fabs(Q[iq])>QI1[0])D[iq]=DI[0];
if(fabs(Q[iq])<QI1[ip]&&fabs(Q[iq])>QI1[ip+1])D[iq]=DI[ip];
if(fabs(Q[iq])<QI1[M3-2])D[iq]=DI[M3-1];
      }
    }
    }
  }
if(Pp==1)printf("管径分配 D[i]:\n");
                for(iq=0;iq<M;iq++)
  {
S[iq]=1.736e-3*L[iq]/pow(D[iq],5.29);
if(Pp==1){    printf("[%3d] %6.4f ",iq,D[iq]);
if(iq%4==3)printf("\n");
    }
  }
if(Pp==1)  printf("\n");
do

 {

for(i=0;i<M;i++)
  {
K3=1.;
```

```
V[i]=fabs(Q[i])*4./(PI*D[i]*D[i]);
if(V[i]<1.0e-4)K3=0.0;
else if(V[i]<1.2)
    {
K3=0.852*pow((1.+0.867/V[i]),0.3);
    }

H[i]=K3*S[i]*fabs(Q[i])*Q[i];
for(j=0;j<=N1;j++)
if(i==qsrc[j].Psrc)H[i]=(H[i]-qsrc[j].Hsrc)*abs(IO[i])/IO[i];

/*      if(KO%5000==4999)if(ps[9]==0)printf("H[%3d] %8.5f  \n",i,H[i]);*/
  }

/*计算水头损失*/
for(i=0;i<N;i++)
  {
F[i]=0;
R[i]=0;
DQ[i]=0.0;

for(k=0;k<M;k++)
    {
if((abs(IO[k])-1)==i)
      {
F[i]=F[i]+H[k];
R[i]=R[i]+S[k]*fabs(Q[k]);
      }
if((abs(JO[k])-1)==i)
      {
F[i]=F[i]-H[k];
R[i]=R[i]+S[k]*fabs(Q[k]);
      }
    }
if(R[i]>1e-5)DQ[i]=-F[i]/R[i]/2.0;
```

```
 if(R[i]<1e-5)DQ[i]=0.0;
 /*计算调整流量*/
   }
 for(k=0;k<M;k++)
   {
 if(JO[k]!=0)
      {
 Q[k]=Q[k]+DQ[abs(IO[k])-1]-DQ[abs(JO[k])-1];
      }
 if(JO[k]==0)
      {
 if(abs(IO[k])<N)Q[k]=Q[k]+DQ[abs(IO[k])-1];
 if(abs(IO[k])>=N)Q[k]=Q[k];
      }
/*if(ps[9]==0)       printf("Q[%d]=%f\n",k,Q[k]);*/
   }    /*计算调整后流量*/
 js=0;
 for(i=0;i<N;i++)
 if(fabs(F[i])>eps)js++;
/*if(ps[9]==0){      printf("-----  %d loops run -----\n",KO);
 printf(" *i*       F            R            DQ\n");
 for(i=0;i<N;i++)
  {
 printf("*%2d*  %12.6f  %12.6f  %12.6f\n",i,F[i],R[i],DQ[i]);
  }
   }*/
 if(js==0&&id==M2-1)
{
 for(i=0;i<M;i++)
  {
 pipe1[0][i]=pipe[0][i];
 pipe1[1][i]=pipe[1][i];
/*       printf("p1[%d]=%d  p2[%d]=%d\n",i,pipe1[0][i],i,pipe1[1][i]);*/
 printf("",pipe1[0][i],pipe1[1][i]);
  }
```

```
 for(i=0;i<M;i++)
  {
 if((Q[i]*(float)(IO[i]))<0.0)
    {
 pipe1[0][i]=pipe[1][i];
 pipe1[1][i]=pipe[0][i];

    }
  }
 /*计算总水头损失*/
 for(i=0;i<M1;i++)Hb[i].jus=0;
/*if(!(ps[8]==0&&ps[7]==1&&is==0))*/for(i=0;i<N1;i++)
  {
 ns=qsrc[i].Nsrc;
/*if(ps[4]==1)*/{Hb[ns].Hfre=qsrc[i].Hsrc;
 Hb[ns].jus=1;
 }

 Hs=0.0;
 iq=0;
/*        pt[0]=M+1;*/
 printf("\n 水流路径  H %3d",ns);
 for(ip=0;ip<M1;ip++)
    {
 for(j=0;j<M;j++)
      {
 if(Pb==1&&j==Pbad)j++;
 for(k=0;k<2;k++)
       {
 if(pipe1[k][j]==ns)
         {
 if(k==0)
           {
 ns=pipe1[1][j];
```

```
Hs+=fabs(H[j]);
printf("-%2d",ns);
          }
        }
      }
    }
  }

printf("\n 总水头损失  Hsum= %8.4f\n",Hs);
 }

for(j=0;j<=N1;j++)
H[qsrc[j].Psrc]=qsrc[j].Hsrc-fabs(H[qsrc[j].Psrc]);
for(ip=0;ip<M1;ip++)
 {
jjt=0;
for(k=0;k<M1;k++)if(Hb[k].jus==0)
  {
jjt=1;
break;
  }
if(jjt==0)break;
for(j=0;j<M;j++)
  {
if(Pb==1&&j==Pbad)j++;
if(Hb[pipe1[0][j]].jus==1)
    {
Hb[pipe1[1][j]].Hfre=fabs(Hb[pipe1[0][j]].Hfre)+Hb[pipe1[0][j]].Habs-
        Hb[pipe1[1][j]].Habs-fabs(H[j]);
Hb[pipe1[1][j]].jus=1;
    }
if(Hb[pipe1[1][j]].jus==1)
    {
Hb[pipe1[0][j]].Hfre=fabs(Hb[pipe1[1][j]].Hfre)+Hb[pipe1[1][j]].Habs
        -Hb[pipe1[0][j]].Habs+fabs(H[j]);
```

```
Hb[pipe1[0][j]].jus=1;
    }
  }
 }


/*计算实际的节点流量*/
for(i=0;i<M1;i++)
 {
q[i]=0.0;
 }
for(i=0;i<M1;i++)
 {
for(j=0;j<M;j++)
  {
for(k=0;k<2;k++)
    {
if(pipe1[k][j]==i)
      {
if(k==0)
        {
q[i]=q[i]-fabs(Q[j]);
if(is==0)q2[i]=q[i];
        }
if(k==1)
        {
q[i]=q[i]+fabs(Q[j]);
if(is==0)q2[i]=q[i];
      }
      }
    }
  }
/*       if(is==0)printf("q[%d]=%f\n",i,q[i]);*/
 }
```

```
        if(!(ps[8]==0&&ps[7]==1&&is==0))
        {
        /*      printf("----- 运行了 %d 循环 -----\n",KO);*/
        printf("管号  水损 水力坡  管长  管径  流量  流速 方向 节点流量 水压 高程  \n");
        printf(" i|  *H*  *1000i* *L*  *D*  *Q*  *V* dire. *q*  *h* * - *  \n");
        printf("------mH2O--mH2O----m-----m-----m3/s--m/s-A ->B -m3/s---m----m---\n");
         for(i=0;i<M;i++)
          {
         if(i<M1)
           {
         if(L[i]>1e-8)printf("%3d| %5.2f %5.2f %6.1f %5.3f %6.4f %4.2f %2d>%2d %6.4f %4.1f %4.1f\n",
         i,fabs(H[i]),fabs(H[i])*1000./L[i],L[i],D[i],fabs(Q[i]),V[i],pipe1[0][i],
         pipe1[1][i],fabs(q[i]),Hb[i].Hfre,Hb[i].Habs);
         if(L[i]<1e-8)printf("%3d| %5.2f  **** %6.1f %5.3f %6.4f %4.2f %2d>%2d %6.4f %4.1f\n",
         i,fabs(H[i]),L[i],D[i],fabs(Q[i]),V[i],pipe1[0][i],pipe1[1][i],fabs(q[i]),
         Hb[i].Hfre,Hb[i].Habs);
           }
         if(i>=M1)
           {
         if(L[i]>1e-8)printf("%3d| %5.2f %5.2f %6.1f %5.3f %6.4f %4.2f %2d>%2d\n",

i,fabs(H[i]),fabs(H[i])*1000./L[i],L[i],D[i],fabs(Q[i]),V[i],pipe1[0][i],pipe1[1][i]);
         if(L[i]<1e-8)printf("%3d| %5.2f  **** %6.1f %5.3f %6.4f %4.2f %2d>%2d\n",
         i,fabs(H[i]),L[i],D[i],fabs(Q[i]),V[i],pipe1[0][i],pipe1[1][i]);
           }
          }

        /*      printf("\n------ Diameter ajusted  %2d time altogather -----\n",id+1);*/
```

```
 printf("\n------ 管径调整共    %2d 次  -----\n",id+1);
}

if(ps[9]==0)if((!(ps[8]==0&&ps[7]==1))&&is==0)
  {
 printf("各节点自由水头:\n");
 for(i=0;i<M1;i++)
    {
 printf(" [%3d] %5.1f ",i,Hb[i].Hfre);
 if(ps[4]==1)printf(" %5.1f ",Hb[i].Habs);
 if(i%4==3)printf("\n");
    }
 printf("\n");
  }

if((!(ps[8]==0&&ps[7]==1))&&is==0)
{
 C=0.0;
 soil=0.0;
 fill=0.0;
 base=0.0;
 pad=0.0;
 for(i=0;i<M;i++)
  {
 soil=soil+(D[i]+2.*wide)*(depth1+D[i]+depth2+depth3)*L[i];
 fill=fill+(D[i]+2.*wide)*(depth1+D[i]/3.)*L[i];
 base=base+D[i]*depth2*L[i];
 pad=pad+(D[i]+2.*wide)*depth3*L[i];
  }
 if(ps[5]==0)
  {
 for(i=0;i<M;i++)
    {
 C=C+12.+372.*pow(D[i],1.7)*L[i];
```

```
    }
  C=C*costb/83.6;
   }
  if(ps[5]==1)
   {
 /*      printf("b=%f  a=%f costb=%f\n",b,a,costb);*/
  for(i=0;i<M;i++)
    {
  C=C+costb1+b*pow(D[i],a)*L[i];
    }
   }      ps[5]=0;

  printf("\n        ***  管网平差成功  ! ***\n\n");
  printf("  ================== 经济参考指数: ==================\n");
  printf("  $$$  管子总造价估算为     %15.2f 元。     $$$\n",C);
  printf("  $$$  工程总挖方估算为     %15.2f 立方。  $$$\n",soil);
  printf("  $$$  回填土方总量估算为  %15.2f 立方。  $$$\n",fill);
  printf("  $$$  基础用材总量估算为  %15.2f 立方。  $$$\n",base);
  printf("  $$$  垫层用材总量估算为  %15.2f 立方。  $$$\n",pad);
  printf("  ================== 经济参考指数: ==================\n");
 }
  if(Px==1)Px=2;
  if(Pz==1)Pz=2;
  if(Pb==1)Pb=2;
  if(ps[6]==1&&Pz==0)Pz=1;
  if(ps[7]==1&&Px==0&&(Pz!=1))Px=1;
  if(ps[8]==1&&Pb==0&&(Pz!=1)&&(Px!=1))Pb=1;
  if(Pz==1)
 {
  printf("\n================== 转输流量校核
==================\n");
  printf("水塔节点号  %3d,转输系数[Qz/Qmax]: %5.3f-->\n",Pt,Xz);
  for(i=0;i<M1;i++)q[i]=q2[i]*Xz;
  Qsum=0.0;
  for(i=0;i<N1;i++)Qsum=Qsum+qsrc[i].Qsrc;
```

```
 q[Pt]=q[Pt]+Qsum*(1.0-Xz);
/*       printf("q[pt] %5.3f Qsum %6.4f\n",q[Pt],Qsum);
 if(ps[9]==0)for(i=0;i<M1;i++)printf("q[%d] %f\n",i,q[i]);*/
 ps[0]=0;
 ps[1]=1;
 ps[2]=0;
 ps[3]=1;
}
 if(ps[6]==1&&Pz==2)
{
 printf("\n       ***  转输流量校核成功   ! ***\n");
 Pz=3;
}

 if(Px==1)
  {
 printf("\n==================== 消防校核  ====================
=\n");
 for(i=0;i<Nf;i++)
  {
/*       scanf("%d",&Nof[i]);*/
 printf(" %d ",Nof[i]);
  }
 printf("节点同时着火,水量  %f * %d m3/s\n",Wf,Nf);
 for(i=0;i<M1;i++)q[i]=q2[i];
 for(i=0;i<Nf;i++)
 q[Nof[i]]=q[Nof[i]]+Wf;
 for(i=0;i<N1;i++)
 qsrc[i].Qsrc=qsrc[i].Qsrc+Wf*(float)(Nf)/(float)(N1);
 ps[0]=0;
 ps[1]=1;
 ps[2]=0;
 ps[3]=1;
if(ps[9]==0)  for(i=0;i<M1;i++)printf("q[%d] %f\n",i,q[i]);
  }
```

```
        if(ps[7]==1&&Px==2)
         {
        printf("\n       ***  消防流量校核成功   ! ***\n");
        Px=3;
         }
        if(Pb==1)
         {
        printf("\n================== 事故流量校核
==================\n");
        for(i=0;i<Nf;i++)
         {
        printf(" %d ",Nof[i]);
         }
       printf("节点同时着火,水量( %6.4f ) * (%1d) m3/s,事故管段
%3d\n",Wf,Nf,Pbad);
        for(i=0;i<M;i++)
          {
        if(i!=Pbad)
           {
        if(abs(IO[i])==abs(IO[Pbad]))
            {
        if(JO[Pbad]!=0)IO[i]=abs(JO[Pbad])*abs(IO[i])/IO[i];
        if(JO[Pbad]==0)IO[i]=N+1;
            }
        if(abs(JO[i])==abs(IO[Pbad]))
            {
        if(JO[Pbad]!=0)JO[i]=abs(JO[Pbad])*abs(JO[i])/JO[i];
        if(JO[Pbad]==0)JO[i]=N+1;
            }
           }
          }

        IO[Pbad]=N+1;
        JO[Pbad]=0;
        for(i=0;i<M1;i++)
```

```
    {
   q[i]=q2[i]*.7;
  /*      for(i=0;i<M1;i++)printf("q[%d] %f\n",i,q[i]);*/
    }

   for(i=0;i<N1;i++)
    {
   if(ps[7]==1)qsrc[i].Qsrc=qsrc[i].Qsrc-Wf*(float)(Nf)/(float)(N1);
   qsrc[i].Qsrc=qsrc[i].Qsrc*.7+Wf*(float)(Nf)/(float)(N1);
    }
   for(i=0;i<Nf;i++)
   q[Nof[i]]=q[Nof[i]]+Wf;
   ps[0]=0;
   ps[1]=1;
   ps[2]=0;
   ps[3]=1;

    }
  }
  if(ps[8]==1&&Pb==2)printf("\n       ***  事故流量校核成功    ! ***\n");

   KO++;
   if(id<M2-1)
   if(KO>5000)
  {
   break;
  }
   if(id==M2-1)if(KO>10000)
  {
  printf("可能管径配合不合适或控制流速,闭合差不适当,计算非正常中止!
\n");
  exit(1);
  }
  }while(js>0);
  }
```

```
}
}

void
on_Cancel_clicked                (GtkButton       *button,
                                  gpointer         user_data)
{
gtk_main_quit();

}


//main.c
//=========================================================
/*
  * Initial main.c file generated by Glade. Edit as required.
  * Glade will not overwrite this file.
  */

#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <gtk/gtk.h>

#include "interface.h"
#include "support.h"

#define Ng 12
int ps[Ng],i; //0605

int
main (int argc, char *argv[])
{
   GtkWidget *window1;
```

```
#ifdef ENABLE_NLS
  bindtextdomain (GETTEXT_PACKAGE, PACKAGE_LOCALE_DIR);
  bind_textdomain_codeset (GETTEXT_PACKAGE, "UTF-8");
  textdomain (GETTEXT_PACKAGE);
#endif

  gtk_set_locale ();
  gtk_init (&argc, &argv);
for(i=0;i<11;i++)ps[i]=0;
ps[2]=1;
ps[5]=1;
ps[10]=1;

  add_pixmap_directory (PACKAGE_DATA_DIR "/" PACKAGE
"/pixmaps");

  /*
   * The following code was added by Glade to create one of each component
   * (except popup menus), just so that you see something after building
   * the project. Delete any components that you don't want shown initially.
   */
  window1 = create_window1 ();
  gtk_widget_show (window1);

  gtk_main ();
  return 0;
}

//interface.c
//=======
/*
 * DO NOT EDIT THIS FILE - it is generated by Glade.
 */
```

```
#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>

#include <gdk/gdkkeysyms.h>
#include <gtk/gtk.h>

#include "callbacks.h"
#include "interface.h"
#include "support.h"

#define GLADE_HOOKUP_OBJECT(component,widget,name) \
  g_object_set_data_full (G_OBJECT (component), name, \
    gtk_widget_ref (widget), (GDestroyNotify) gtk_widget_unref)

#define GLADE_HOOKUP_OBJECT_NO_REF(component,widget,name) \
  g_object_set_data (G_OBJECT (component), name, widget)

GtkWidget*
create_window1 (void)
{
  GtkWidget *window1;
  GtkWidget *vbox1;
  GtkWidget *notebook1;
  GtkWidget *scrolledwindow2;
  GtkWidget *textview1;
  GtkWidget *label1;
  GtkWidget *vbox6;
  GtkWidget *frame5;
```

```
GtkWidget *alignment5;
GtkWidget *vbox7;
GtkWidget *frame2;
GtkWidget *alignment2;
GtkWidget *vbox3;
GtkWidget *radiobutton1;
GSList *radiobutton1_group = NULL;
GtkWidget *radiobutton2;
GtkWidget *radiobutton3;
GtkWidget *label3;
GtkWidget *checkbutton1;
GtkWidget *frame3;
GtkWidget *alignment3;
GtkWidget *vbox4;
GtkWidget *radiobutton6;
GSList *radiobutton6_group = NULL;
GtkWidget *radiobutton7;
GtkWidget *label4;
GtkWidget *label6;
GtkWidget *hseparator3;
GtkWidget *frame4;
GtkWidget *alignment4;
GtkWidget *vbox5;
GtkWidget *T_CK;
GtkWidget *F_CK;
GtkWidget *A_CK;
GtkWidget *label5;
GtkWidget *hseparator4;
GtkWidget *frame1;
GtkWidget *alignment1;
GtkWidget *vbox2;
GtkWidget *checkbutton2;
GtkWidget *checkbutton3;
GtkWidget *label2;
```

```
	GtkWidget *fixed4;
	GtkWidget *hbox1;
	GtkWidget *fixed1;
	GtkWidget *Ok;
	GtkWidget *fixed2;
	GtkWidget *Cancel;
	GtkWidget *fixed3;

	window1 = gtk_window_new (GTK_WINDOW_TOPLEVEL);
	gtk_window_set_title (GTK_WINDOW (window1),
_("\350\257\267\351\200\211\346\213\251\347\273\231\346\260\264\347\256\241\3
47\275\221\345\271\263\345\267\256\345\217\202\346\225\260"));

	vbox1 = gtk_vbox_new (FALSE, 0);
	gtk_widget_show (vbox1);
	gtk_container_add (GTK_CONTAINER (window1), vbox1);

	notebook1 = gtk_notebook_new ();
	gtk_widget_show (notebook1);
	gtk_box_pack_start (GTK_BOX (vbox1), notebook1, TRUE, TRUE, 0);

	scrolledwindow2 = gtk_scrolled_window_new (NULL, NULL);
	gtk_widget_show (scrolledwindow2);
	gtk_container_add (GTK_CONTAINER (notebook1), scrolledwindow2);
	gtk_widget_set_size_request (scrolledwindow2, 500, 150);
	gtk_scrolled_window_set_shadow_type (GTK_SCROLLED_WINDOW
(scrolledwindow2), GTK_SHADOW_IN);

	textview1 = gtk_text_view_new ();
	gtk_widget_show (textview1);
	gtk_container_add (GTK_CONTAINER (scrolledwindow2), textview1);
	gtk_text_view_set_editable (GTK_TEXT_VIEW (textview1), FALSE);
	gtk_text_buffer_set_text (gtk_text_view_get_buffer (GTK_TEXT_VIEW
(textview1)),
```

_("\342\200\234\345\271\263\345\267\256\344\271\213\346\230\237\342\200\235\350\275\257\344\273\266,\347\211\210\346\235\203\346\211\200\346\234\211(C)2006<\344\275\234\350\200\205\357\274\232\346\226\275\345\273\272\347\272\262>sjg255@sh163d.sta.net.cn\n\346\254\242\350\277\216\350\256\277\351\227\256http://www.egesh.com\350\216\267\345\217\226\346\234\200\346\226\260\344\277\241\346\201\257;\344\270\272\344\276\277\344\272\216\n\346\202\250\346\233\264\345\245\275\345\234\260\344\275\277\347\224\250\357\274\214\350\257\267\345\217\202\351\230\205\344\270\213\351\235\242\347\232\204\347\232\204\344\275\277\347\224\250\350\257\264\346\230\216\345\222\214\344\275\277\347\224\250\346\211\213\345\206\214;\n\346\234\254\350\275\257\344\273\266\346\230\257\350\207\252\347\224\261\350\275\257\344\273\266\357\274\214\344\275\240\345\217\257\344\273\245\351\201\265\347\205\247\350\207\252\347\224\261\350\275\257\344\273\266\345\237\272\351\207\221\344\274\232(FSF)\n\345\207\272\347\211\210\347\232\204GNU\351\200\232\347\224\250\345\205\254\345\205\261\350\256\270\345\217\257\350\257\201(GPL)\346\235\241\346\254\276\346\235\245\344\277\256\346\224\271\345\222\214\351\207\215\346\226\260\n\345\217\221\345\270\203\350\277\231\344\270\200\347\250\213\345\272\217\343\200\202\n\345\217\221\345\270\203\350\277\231\344\270\200\347\250\213\345\272\217\347\232\204\347\233\256\347\232\204\346\230\257\345\270\214\346\234\233\345\256\203\346\234\211\347\224\250\357\274\214\344\275\206\346\262\241\346\234\211\344\273\273\344\275\225\346\213\205\344\277\235\343\200\202\n\347\224\232\350\207\263\346\262\241\346\234\211\351\200\202\345\220\210\347\211\271\345\256\232\347\233\256\347\232\204\347\232\204\351\232\220\345\220\253\347\232\204\346\213\205\344\277\235\343\200\202\346\233\264\350\257\246\347\273\206\347\232\204\346\203\205\345\206\265\n\350\257\267\345\217\202\351\230\205GNU\351\200\232\347\224\250\345\205\254\345\205\261\350\256\270\345\217\257\350\257\201\343\200\202\n\n\346\234\254\350\275\257\344\273\266\346\230\257\345\270\202\346\224\277\347\273\231\346\260\264\344\270\223\344\270\232\350\256\241\347\256\227\350\275\257\344\273\266\357\274\214\344\270\272\346\202\250\346\217\220\344\276\233\344\272\206\345\233\233\347\247\215\346\265\201\351\207\217\n\347\232\204\350\276\223\345\205\245\346\263\225;
\344\270\244\347\247\215\347\241\256\345\256\232\347\256\241\345\276\204\347\232\204\346\226\271\346\263\225\345\217\212\350\275\254\350\276\223.\346\266\210\351\230\262\345\222\214\344\272\213\346\225\205\346\240\241\346\240\270.\n\344\270\272\344\277\235\350\257\201\346\202\250\347\232\204\350\256\241\347

\256\227\345\234\206\346\273\241\346\210\220\345\212\237\357\274\214\345\273\
272\350\256\256\345\205\210\347\273\230\345\210\266\350\256\241\347\256\227\3
47\256\200\345\233\276\357\274\214\347\204\266\345\220\216\n\346\214\211\344\
273\216\345\267\246\345\210\260\345\217\263\357\274\214\344\273\216\344\270\2
12\345\210\260\344\270\213\347\232\204\351\241\272\345\272\217\347\274\226\34
5\210\266\350\212\202\347\202\271\345\217\267\343\200\201\347\216\257\345\217
\267\357\274\214\n\350\212\202\347\202\271\345\217\267\345\222\214\347\216\25
7\345\217\267\346\225\260\345\200\274\344\273\2161\345\274\200\345\247\213\34
7\274\226\345\210\266\357\274\2140\350\212\202\347\202\271\344\277\235\347\22
5\231\344\270\272\346\260\264\346\272\220\350\212\202\347\202\271\345\217\267
;\n\345\246\202\346\236\234\346\230\257\345\244\232\346\260\264\346\272\220\35
7\274\2140\350\212\202\347\202\271\345\222\214\346\260\264\346\272\220\346\26
5\201\345\205\245\347\232\204\345\220\204\347\202\271\347\273\204\346\210\220
\350\231\232\347\216\257;\n\347\256\241\346\256\265\345\217\267\346\225\260\34
5\200\274\344\273\2160\345\274\200\345\247\213\347\274\226\345\210\266;\n\345\
220\204\347\256\241\346\256\265\347\232\204\347\216\257\345\217\267\351\241\2
72\346\227\266\351\222\210\344\270\272\346\255\243\357\274\214\351\200\206\34
6\227\266\351\222\210\344\270\272\350\264\237;\n\346\225\260\346\215\256\346\2
26\207\344\273\266\347\232\204\347\274\226\345\210\266\345\217\257\345\234\25
0\347\241\256\345\256\232\n\346\260\264\346\272\220\346\225\260(N1),\347\256\
241\346\225\260(M),\347\216\257\346\225\260(N),\350\212\202\347\202\271\346\2
25\260(M1),\350\260\203\346\225\264\347\256\241\345\276\204\n\347\232\204\351
\201\215\346\225\260(M2)\345\220\216\n\351\200\211\346\213\251\344\270\213\35
1\235\242\347\232\204[Help--
\346\225\260\346\215\256\346\226\207\344\273\266\350\276\223\345\205\245\345\
270\256\345\212\251]\345\220\216\346\214\211\350\276\223\345\207\272\347\232\
204\347\273\223\346\236\234\346\225\264\347\220\206\343\200\202\n\n\350\256\2
41\347\256\227\345\260\206\350\276\223\345\207\272\346\265\201\345\220\221,\3
46\265\201\351\200\237,\350\212\202\347\202\271\346\265\201\351\207\217,\350\2
12\202\347\202\271\346\260\264\345\244\264,\347\256\241\351\201\223\346\265\2
01\351\207\217,\n\347\256\241\345\276\204,\346\260\264\346\215\237\345\217\212
\347\256\241\351\201\223\344\273\267\346\240\274\347\232\204\344\274\260\347\
256\227.\n\344\270\213\351\235\242\345\220\204\351\241\271\345\210\206\345\21
0\253\350\241\250\347\244\272: \n\357\274\2101\357\274\211Qpipe--
\347\256\241\346\256\265\346\265\201\351\207\217\350\276\223\345\205\245

;\n\357\274\2102\357\274\211Qnode--
\350\212\202\347\202\271\346\265\201\351\207\217\350\276\223\345\205\245
;\n\357\274\2103\357\274\211Qc_node--
\351\233\206\344\270\255\346\265\201\351\207\217\350\276\223\345\205\245\n\n\3
57\274\2104\357\274\211Height--
\346\230\257\345\220\246\350\276\223\345\205\245\350\212\202\347\202\271\351\
253\230\347\250\213;\n\n\357\274\2105\357\274\211Diameter--
\346\230\257\345\220\246\346\234\211\347\241\256\345\256\232\347\232\204\347\
256\241\345\276\204\350\276\223\345\205\245 ;\n
\357\274\210\347\224\250\344\272\216\345\267\262\346\234\211\347\256\241\347\
275\221\346\240\241\346\240\270\357\274\214\346\210\226\346\211\251\345\273\2
72\347\256\241\347\275\221\357\274\214\346\211\251\345\273\272\347\256\241\34
7\275\221\344\270\255\346\234\252\345\256\232\347\256\241\345\276\204\350\276
\2230.0\357\274\211\n\357\274\2106\357\274\211Eco.-Tech.--
\346\214\211\347\273\217\346\265\216(\346\210\226\346\216\247\345\210\266)\34
6\265\201\351\200\237\351\200\211\347\256\241\345\276\204\345\271\266\344\274
\260\347\256\227;\n
\357\274\210\347\224\250\344\272\216\346\262\241\346\234\211\347\241\256\345\
256\232\347\232\204\347\256\241\345\276\204\346\227\266\357\274\214\345\246\2
02\346\226\260\345\273\272\347\256\241\347\275\221\357\274\211\n\n\357\274\21
07\357\274\211Trans.-CK--
\350\275\254\350\276\223\346\265\201\351\207\217\346\240\241\346\240\270;\n\35
7\274\2108\357\274\211Fire-CK--
\346\266\210\351\230\262\346\265\201\351\207\217\346\240\241\346\240\270;\n\35
7\274\2109\357\274\211Accid.-CK--
\344\272\213\346\225\205\346\265\201\351\207\217\346\240\241\346\240\270
;\n\357\274\21010\357\274\211Output--
\347\262\276\347\256\200\350\276\223\345\207\272
;\n\357\274\21011\357\274\211\346\225\260\346\215\256\350\276\223\345\205\245\
345\270\256\345\212\251;
\n\357\274\2101\357\274\211\343\200\201\357\274\2102\357\274\211\343\200\201\
357\274\2103\357\274\211\351\241\271\344\273\273\351\200\211\344\270\200;\n\3
57\274\2105\357\274\211\343\200\201\357\274\2106\357\274\211\344\272\214\351\
200\211\344\270\200;\n\351\200\211\357\274\21010\357\274\211\345\217\252\350\
276\223\345\207\272\347\273\223\346\236\234\357\274\214\345\220\246\345\210\2

```
31\345\260\206\350\276\223\345\207\272\345\276\210\350\257\246\347\273\206\34
7\232\204\344\270\255\351\227\264\350\256\241\347\256\227\350\277\207\347\250
\213;\n\351\200\211\357\274\21011\357\274\211\345\217\252\350\276\223\345\207\
272\345\270\256\345\212\251\357\274\214\345\220\246\345\210\231\345\260\206\3
50\277\233\350\241\214\350\256\241\347\256\227\343\200\202\n\350\257\267\351\
200\211\346\213\251[Ok]\351\224\256\347\273\247\347\273\255\346\210\226[Canc
el]\351\224\256\351\200\200\345\207\272\n\n\347\245\235\346\202\250\344\275\27
7\347\224\250\346\204\211\345\277\253\357\274\201\n"), -1);

        label1 = gtk_label_new (_("\350\257\264\346\230\216"));
        gtk_widget_show (label1);
        gtk_notebook_set_tab_label (GTK_NOTEBOOK (notebook1),
gtk_notebook_get_nth_page (GTK_NOTEBOOK (notebook1), 0), label1);

        vbox6 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox6);
        gtk_box_pack_start (GTK_BOX (vbox1), vbox6, TRUE, TRUE, 0);

        frame5 = gtk_frame_new (NULL);
        gtk_widget_show (frame5);
        gtk_box_pack_start (GTK_BOX (vbox6), frame5, TRUE, TRUE, 0);
        gtk_frame_set_shadow_type (GTK_FRAME (frame5),
GTK_SHADOW_ETCHED_OUT);

        alignment5 = gtk_alignment_new (0.5, 0.5, 1, 1);
        gtk_widget_show (alignment5);
        gtk_container_add (GTK_CONTAINER (frame5), alignment5);
        gtk_alignment_set_padding (GTK_ALIGNMENT (alignment5), 0, 0, 12, 0);

        vbox7 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox7);
        gtk_container_add (GTK_CONTAINER (alignment5), vbox7);

        frame2 = gtk_frame_new (NULL);
```

```
        gtk_widget_show (frame2);
        gtk_box_pack_start (GTK_BOX (vbox7), frame2, TRUE, TRUE, 0);

        alignment2 = gtk_alignment_new (0.5, 0.5, 1, 1);
        gtk_widget_show (alignment2);
        gtk_container_add (GTK_CONTAINER (frame2), alignment2);
        gtk_alignment_set_padding (GTK_ALIGNMENT (alignment2), 0, 0, 12, 0);

        vbox3 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox3);
        gtk_container_add (GTK_CONTAINER (alignment2), vbox3);

        radiobutton1 = gtk_radio_button_new_with_mnemonic (NULL, _("Qpipe-
\347\256\241\346\256\265\346\265\201\351\207\217\350\276\223\345\205\245\n"));
        gtk_widget_show (radiobutton1);
        gtk_box_pack_start (GTK_BOX (vbox3), radiobutton1, FALSE, FALSE, 0);
        gtk_radio_button_set_group (GTK_RADIO_BUTTON (radiobutton1),
radiobutton1_group);
        radiobutton1_group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (radiobutton1));

        radiobutton2 = gtk_radio_button_new_with_mnemonic (NULL, _("Qnode-
\350\212\202\347\202\271\346\265\201\351\207\217\350\276\223\345\205\245"));
        gtk_widget_show (radiobutton2);
        gtk_box_pack_start (GTK_BOX (vbox3), radiobutton2, FALSE, FALSE, 0);
        gtk_radio_button_set_group (GTK_RADIO_BUTTON (radiobutton2),
radiobutton1_group);
        radiobutton1_group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (radiobutton2));

        radiobutton3 = gtk_radio_button_new_with_mnemonic (NULL, _("Qc_node-
\351\233\206\344\270\255\346\265\201\351\207\217\350\276\223\345\205\245"));
        gtk_widget_show (radiobutton3);
        gtk_box_pack_start (GTK_BOX (vbox3), radiobutton3, FALSE, FALSE, 0);
```

```
        gtk_radio_button_set_group (GTK_RADIO_BUTTON (radiobutton3),
radiobutton1_group);
        radiobutton1_group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (radiobutton3));
        gtk_toggle_button_set_active (GTK_TOGGLE_BUTTON (radiobutton3),
TRUE);

        label3 = gtk_label_new
(_("<b>\346\265\201\351\207\217\350\276\223\345\205\245\351\200\211\346\213\2
51</b>"));
        gtk_widget_show (label3);
        gtk_frame_set_label_widget (GTK_FRAME (frame2), label3);
        gtk_label_set_use_markup (GTK_LABEL (label3), TRUE);

        checkbutton1 = gtk_check_button_new_with_mnemonic (_("Height--
\350\276\223\345\205\245\350\212\202\347\202\271\351\253\230\347\250\213"));
        gtk_widget_show (checkbutton1);
        gtk_box_pack_start (GTK_BOX (vbox7), checkbutton1, FALSE, FALSE, 0);

        frame3 = gtk_frame_new (NULL);
        gtk_widget_show (frame3);
        gtk_box_pack_start (GTK_BOX (vbox7), frame3, FALSE, FALSE, 0);

        alignment3 = gtk_alignment_new (0.5, 0.5, 1, 1);
        gtk_widget_show (alignment3);
        gtk_container_add (GTK_CONTAINER (frame3), alignment3);
        gtk_alignment_set_padding (GTK_ALIGNMENT (alignment3), 0, 0, 12, 0);

        vbox4 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox4);
        gtk_container_add (GTK_CONTAINER (alignment3), vbox4);

        radiobutton6 = gtk_radio_button_new_with_mnemonic (NULL,
_("Diameter--
```

```
\346\234\211\347\241\256\345\256\232\347\232\204\347\256\241\345\276\204\350\
276\223\345\205\245"));
        gtk_widget_show (radiobutton6);
        gtk_box_pack_start (GTK_BOX (vbox4), radiobutton6, FALSE, FALSE, 0);
        gtk_radio_button_set_group (GTK_RADIO_BUTTON (radiobutton6),
radiobutton6_group);
        radiobutton6_group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (radiobutton6));

        radiobutton7 = gtk_radio_button_new_with_mnemonic (NULL, _("Eco.-
Tech.--
\346\214\211\347\273\217\346\265\216(\346\210\226\346\216\247\345\210\266)\34
6\265\201\351\200\237\351\200\211\347\256\241\345\276\204\345\271\266\344\274
\260\347\256\227"));
        gtk_widget_show (radiobutton7);
        gtk_box_pack_start (GTK_BOX (vbox4), radiobutton7, FALSE, FALSE, 0);
        gtk_radio_button_set_group (GTK_RADIO_BUTTON (radiobutton7),
radiobutton6_group);
        radiobutton6_group = gtk_radio_button_get_group
(GTK_RADIO_BUTTON (radiobutton7));
        gtk_toggle_button_set_active (GTK_TOGGLE_BUTTON (radiobutton7),
TRUE);

        label4 = gtk_label_new
(_("<b>\347\256\241\345\276\204\350\276\223\345\205\245\351\200\211\346\213\2
51</b>"));
        gtk_widget_show (label4);
        gtk_frame_set_label_widget (GTK_FRAME (frame3), label4);
        gtk_label_set_use_markup (GTK_LABEL (label4), TRUE);

        label6 = gtk_label_new
(_("<b>\350\276\223\345\205\245\351\200\211\346\213\251</b>"));
        gtk_widget_show (label6);
        gtk_frame_set_label_widget (GTK_FRAME (frame5), label6);
```

```
        gtk_label_set_use_markup (GTK_LABEL (label6), TRUE);

        hseparator3 = gtk_hseparator_new ();
        gtk_widget_show (hseparator3);
        gtk_box_pack_start (GTK_BOX (vbox1), hseparator3, TRUE, TRUE, 0);

        frame4 = gtk_frame_new (NULL);
        gtk_widget_show (frame4);
        gtk_box_pack_start (GTK_BOX (vbox1), frame4, TRUE, TRUE, 0);
        gtk_frame_set_shadow_type (GTK_FRAME (frame4),
GTK_SHADOW_ETCHED_OUT);

        alignment4 = gtk_alignment_new (0.5, 0.5, 1, 1);
        gtk_widget_show (alignment4);
        gtk_container_add (GTK_CONTAINER (frame4), alignment4);
        gtk_alignment_set_padding (GTK_ALIGNMENT (alignment4), 0, 0, 12, 0);

        vbox5 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox5);
        gtk_container_add (GTK_CONTAINER (alignment4), vbox5);

        T_CK = gtk_check_button_new_with_mnemonic
(_("\350\275\254\350\276\223\346\265\201\351\207\217\346\240\241\346\240\270
T-CK"));
        gtk_widget_show (T_CK);
        gtk_box_pack_start (GTK_BOX (vbox5), T_CK, FALSE, FALSE, 0);

        F_CK = gtk_check_button_new_with_mnemonic
(_("\346\266\210\351\230\262\346\265\201\351\207\217\346\240\241\346\240\270
F-CK"));
        gtk_widget_show (F_CK);
        gtk_box_pack_start (GTK_BOX (vbox5), F_CK, FALSE, FALSE, 0);

        A_CK = gtk_check_button_new_with_mnemonic
```

```
(_("\344\272\213\346\225\205\346\265\201\351\207\217\346\240\241\346\240\270
A-CK"));
        gtk_widget_show (A_CK);
        gtk_box_pack_start (GTK_BOX (vbox5), A_CK, FALSE, FALSE, 0);

        label5 = gtk_label_new
(_("<b>\346\240\241\346\240\270\351\200\211\346\213\251</b>"));
        gtk_widget_show (label5);
        gtk_frame_set_label_widget (GTK_FRAME (frame4), label5);
        gtk_label_set_use_markup (GTK_LABEL (label5), TRUE);

        hseparator4 = gtk_hseparator_new ();
        gtk_widget_show (hseparator4);
        gtk_box_pack_start (GTK_BOX (vbox1), hseparator4, TRUE, TRUE, 0);

        frame1 = gtk_frame_new (NULL);
        gtk_widget_show (frame1);
        gtk_box_pack_start (GTK_BOX (vbox1), frame1, TRUE, TRUE, 0);
        gtk_frame_set_shadow_type (GTK_FRAME (frame1),
GTK_SHADOW_ETCHED_OUT);

        alignment1 = gtk_alignment_new (0.5, 0.5, 1, 1);
        gtk_widget_show (alignment1);
        gtk_container_add (GTK_CONTAINER (frame1), alignment1);
        gtk_alignment_set_padding (GTK_ALIGNMENT (alignment1), 0, 0, 12, 0);

        vbox2 = gtk_vbox_new (FALSE, 0);
        gtk_widget_show (vbox2);
        gtk_container_add (GTK_CONTAINER (alignment1), vbox2);

        checkbutton2 = gtk_check_button_new_with_mnemonic (_("Output--
\347\262\276\347\256\200\350\276\223\345\207\272"));
        gtk_widget_show (checkbutton2);
        gtk_box_pack_start (GTK_BOX (vbox2), checkbutton2, FALSE, FALSE, 0);
```

```
        checkbutton3 = gtk_check_button_new_with_mnemonic (_("Help--
\346\225\260\346\215\256\346\226\207\344\273\266\350\276\223\345\205\245\345\
270\256\345\212\251"));
        gtk_widget_show (checkbutton3);
        gtk_box_pack_start (GTK_BOX (vbox2), checkbutton3, FALSE, FALSE, 0);
        gtk_toggle_button_set_active (GTK_TOGGLE_BUTTON (checkbutton3),
TRUE);

        label2 = gtk_label_new
(_("<b>\350\276\223\345\207\272\351\200\211\346\213\251</b>"));
        gtk_widget_show (label2);
        gtk_frame_set_label_widget (GTK_FRAME (frame1), label2);
        gtk_label_set_use_markup (GTK_LABEL (label2), TRUE);

        fixed4 = gtk_fixed_new ();
        gtk_widget_show (fixed4);
        gtk_box_pack_start (GTK_BOX (vbox1), fixed4, TRUE, TRUE, 0);

        hbox1 = gtk_hbox_new (FALSE, 0);
        gtk_widget_show (hbox1);
        gtk_box_pack_start (GTK_BOX (vbox1), hbox1, FALSE, FALSE, 0);

        fixed1 = gtk_fixed_new ();
        gtk_widget_show (fixed1);
        gtk_box_pack_start (GTK_BOX (hbox1), fixed1, TRUE, TRUE, 0);

        Ok = gtk_button_new_with_mnemonic (_("Ok"));
        gtk_widget_show (Ok);
        gtk_box_pack_start (GTK_BOX (hbox1), Ok, FALSE, FALSE, 0);
        gtk_widget_set_size_request (Ok, 80, 27);

        fixed2 = gtk_fixed_new ();
        gtk_widget_show (fixed2);
```

```
gtk_box_pack_start (GTK_BOX (hbox1), fixed2, TRUE, TRUE, 0);

Cancel = gtk_button_new_with_mnemonic (_("Cancel"));
gtk_widget_show (Cancel);
gtk_box_pack_start (GTK_BOX (hbox1), Cancel, FALSE, FALSE, 0);
gtk_widget_set_size_request (Cancel, 80, -1);

fixed3 = gtk_fixed_new ();
gtk_widget_show (fixed3);
gtk_box_pack_start (GTK_BOX (hbox1), fixed3, TRUE, TRUE, 0);

g_signal_connect ((gpointer) radiobutton1, "toggled",
                  G_CALLBACK (Qpipe_toggled),
                  NULL);
g_signal_connect ((gpointer) radiobutton2, "toggled",
                  G_CALLBACK (Qnode_toggled),
                  NULL);
g_signal_connect ((gpointer) radiobutton3, "toggled",
                  G_CALLBACK (Qc_node_toggled),
                  NULL);
g_signal_connect ((gpointer) checkbutton1, "toggled",
                  G_CALLBACK (Height_toggled),
                  NULL);
g_signal_connect ((gpointer) radiobutton6, "toggled",
                  G_CALLBACK (Dimeter_toggled),
                  NULL);
g_signal_connect ((gpointer) radiobutton7, "toggled",
                  G_CALLBACK (Eco__Tec__toggled),
                  NULL);
g_signal_connect ((gpointer) T_CK, "toggled",
                  G_CALLBACK (T_CK_toggled),
                  NULL);
g_signal_connect ((gpointer) F_CK, "toggled",
                  G_CALLBACK (F_CK_toggled),
```

```
                                NULL);
        g_signal_connect ((gpointer) A_CK, "toggled",
                                G_CALLBACK (A_CK_toggled),
                                NULL);
        g_signal_connect ((gpointer) checkbutton2, "toggled",
                                G_CALLBACK (Output_toggled),
                                NULL);
        g_signal_connect ((gpointer) checkbutton3, "toggled",
                                G_CALLBACK (Input_help_toggled),
                                NULL);
        g_signal_connect ((gpointer) Ok, "clicked",
                                G_CALLBACK (on_Ok_clicked),
                                NULL);
        g_signal_connect ((gpointer) Cancel, "clicked",
                                G_CALLBACK (on_Cancel_clicked),
                                NULL);

        /* Store pointers to all widgets, for use by lookup_widget(). */
        GLADE_HOOKUP_OBJECT_NO_REF (window1, window1, "window1");
        GLADE_HOOKUP_OBJECT (window1, vbox1, "vbox1");
        GLADE_HOOKUP_OBJECT (window1, notebook1, "notebook1");
        GLADE_HOOKUP_OBJECT (window1, scrolledwindow2,
"scrolledwindow2");
        GLADE_HOOKUP_OBJECT (window1, textview1, "textview1");
        GLADE_HOOKUP_OBJECT (window1, label1, "label1");
        GLADE_HOOKUP_OBJECT (window1, vbox6, "vbox6");
        GLADE_HOOKUP_OBJECT (window1, frame5, "frame5");
        GLADE_HOOKUP_OBJECT (window1, alignment5, "alignment5");
        GLADE_HOOKUP_OBJECT (window1, vbox7, "vbox7");
        GLADE_HOOKUP_OBJECT (window1, frame2, "frame2");
        GLADE_HOOKUP_OBJECT (window1, alignment2, "alignment2");
        GLADE_HOOKUP_OBJECT (window1, vbox3, "vbox3");
        GLADE_HOOKUP_OBJECT (window1, radiobutton1, "radiobutton1");
        GLADE_HOOKUP_OBJECT (window1, radiobutton2, "radiobutton2");
```

```
GLADE_HOOKUP_OBJECT (window1, radiobutton3, "radiobutton3");
GLADE_HOOKUP_OBJECT (window1, label3, "label3");
GLADE_HOOKUP_OBJECT (window1, checkbutton1, "checkbutton1");
GLADE_HOOKUP_OBJECT (window1, frame3, "frame3");
GLADE_HOOKUP_OBJECT (window1, alignment3, "alignment3");
GLADE_HOOKUP_OBJECT (window1, vbox4, "vbox4");
GLADE_HOOKUP_OBJECT (window1, radiobutton6, "radiobutton6");
GLADE_HOOKUP_OBJECT (window1, radiobutton7, "radiobutton7");
GLADE_HOOKUP_OBJECT (window1, label4, "label4");
GLADE_HOOKUP_OBJECT (window1, label6, "label6");
GLADE_HOOKUP_OBJECT (window1, hseparator3, "hseparator3");
GLADE_HOOKUP_OBJECT (window1, frame4, "frame4");
GLADE_HOOKUP_OBJECT (window1, alignment4, "alignment4");
GLADE_HOOKUP_OBJECT (window1, vbox5, "vbox5");
GLADE_HOOKUP_OBJECT (window1, T_CK, "T_CK");
GLADE_HOOKUP_OBJECT (window1, F_CK, "F_CK");
GLADE_HOOKUP_OBJECT (window1, A_CK, "A_CK");
GLADE_HOOKUP_OBJECT (window1, label5, "label5");
GLADE_HOOKUP_OBJECT (window1, hseparator4, "hseparator4");
GLADE_HOOKUP_OBJECT (window1, frame1, "frame1");
GLADE_HOOKUP_OBJECT (window1, alignment1, "alignment1");
GLADE_HOOKUP_OBJECT (window1, vbox2, "vbox2");
GLADE_HOOKUP_OBJECT (window1, checkbutton2, "checkbutton2");
GLADE_HOOKUP_OBJECT (window1, checkbutton3, "checkbutton3");
GLADE_HOOKUP_OBJECT (window1, label2, "label2");
GLADE_HOOKUP_OBJECT (window1, fixed4, "fixed4");
GLADE_HOOKUP_OBJECT (window1, hbox1, "hbox1");
GLADE_HOOKUP_OBJECT (window1, fixed1, "fixed1");
GLADE_HOOKUP_OBJECT (window1, Ok, "Ok");
GLADE_HOOKUP_OBJECT (window1, fixed2, "fixed2");
GLADE_HOOKUP_OBJECT (window1, Cancel, "Cancel");
GLADE_HOOKUP_OBJECT (window1, fixed3, "fixed3");

return window1;
```

```
}

//support.c
//=======================================================
/*
 * DO NOT EDIT THIS FILE - it is generated by Glade.
 */

#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <sys/types.h>
#include <sys/stat.h>
#include <unistd.h>
#include <string.h>
#include <stdio.h>

#include <gtk/gtk.h>

#include "support.h"

GtkWidget*
lookup_widget                   (GtkWidget       *widget,
                                 const gchar     *widget_name)
{
   GtkWidget *parent, *found_widget;

   for (;;)
      {
        if (GTK_IS_MENU (widget))
          parent = gtk_menu_get_attach_widget (GTK_MENU (widget));
        else
          parent = widget->parent;
        if (!parent)
```

```
            parent = (GtkWidget*) g_object_get_data (G_OBJECT (widget),
"GladeParentKey");
            if (parent == NULL)
              break;
            widget = parent;
          }

        found_widget = (GtkWidget*) g_object_get_data (G_OBJECT (widget),
                                                       widget_name);
        if (!found_widget)
          g_warning ("Widget not found: %s", widget_name);
        return found_widget;
      }

      static GList *pixmaps_directories = NULL;

      /* Use this function to set the directory containing installed pixmaps. */
      void
      add_pixmap_directory                (const gchar     *directory)
      {
        pixmaps_directories = g_list_prepend (pixmaps_directories,
                                              g_strdup (directory));
      }

      /* This is an internally used function to find pixmap files. */
      static gchar*
      find_pixmap_file                    (const gchar     *filename)
      {
        GList *elem;

        /* We step through each of the pixmaps directory to find it. */
        elem = pixmaps_directories;
        while (elem)
          {
```

```
            gchar *pathname = g_strdup_printf ("%s%s%s", (gchar*)elem->data,
                                                G_DIR_SEPARATOR_S,
filename);
            if (g_file_test (pathname, G_FILE_TEST_EXISTS))
               return pathname;
            g_free (pathname);
            elem = elem->next;
         }
      return NULL;
   }

   /* This is an internally used function to create pixmaps. */
   GtkWidget*
   create_pixmap                  (GtkWidget       *widget,
                                   const gchar     *filename)
   {
      gchar *pathname = NULL;
      GtkWidget *pixmap;

      if (!filename || !filename[0])
            return gtk_image_new ();

      pathname = find_pixmap_file (filename);

      if (!pathname)
         {
            g_warning (_("Couldn't find pixmap file: %s"), filename);
            return gtk_image_new ();
         }

      pixmap = gtk_image_new_from_file (pathname);
      g_free (pathname);
      return pixmap;
   }
```

```
/* This is an internally used function to create pixmaps. */
GdkPixbuf*
create_pixbuf                   (const gchar     *filename)
{
   gchar *pathname = NULL;
   GdkPixbuf *pixbuf;
   GError *error = NULL;

   if (!filename || !filename[0])
         return NULL;

   pathname = find_pixmap_file (filename);

   if (!pathname)
      {
         g_warning (_("Couldn't find pixmap file: %s"), filename);
         return NULL;
      }

   pixbuf = gdk_pixbuf_new_from_file (pathname, &error);
   if (!pixbuf)
      {
         fprintf (stderr, "Failed to load pixbuf file: %s: %s\n",
                  pathname, error->message);
         g_error_free (error);
      }
   g_free (pathname);
   return pixbuf;
}

/* This is used to set ATK action descriptions. */
void
glade_set_atk_action_description       (AtkAction       *action,
```

```
                                                const gchar     *action_name,
                                                const gchar     *description)
{
   gint n_actions, i;

   n_actions = atk_action_get_n_actions (action);
   for (i = 0; i < n_actions; i++)
      {
         if (!strcmp (atk_action_get_name (action, i), action_name))
            atk_action_set_description (action, i, description);
      }
}

//interface.h
//=====================================================
/*
 * DO NOT EDIT THIS FILE - it is generated by Glade.
 */

GtkWidget* create_window1 (void);

//callbacks.h
//=====================================================
#include <gtk/gtk.h>


void
Qpipe_toggled                   (GtkToggleButton *togglebutton,
                                                gpointer        user_data);

void
Qnode_toggled                   (GtkToggleButton *togglebutton,
                                                gpointer        user_data);

void
```

```
Qc_node_toggled                 (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
Dimeter_toggled                 (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
Height_toggled                  (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
Eco__Tec__toggled               (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
T_CK_toggled                    (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
F_CK_toggled                    (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
A_CK_toggled                    (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
Output_toggled                  (GtkToggleButton *togglebutton,
                                        gpointer         user_data);

void
Input_help_toggled              (GtkToggleButton *togglebutton,
                                        gpointer         user_data);
```

```
void
on_Ok_clicked                   (GtkButton      *button,
                                        gpointer        user_data);

void
on_Cancel_clicked               (GtkButton      *button,
                                        gpointer        user_data);

void
on_Ok_clicked                   (GtkButton      *button,
                                        gpointer        user_data);

void
on_Cancel_clicked               (GtkButton      *button,
                                        gpointer        user_data);

void
Qpipe_toggled                   (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Qnode_toggled                   (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Qc_node_toggled                 (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Output_toggled                  (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Input_help_toggled              (GtkToggleButton *togglebutton,
                                        gpointer        user_data);
```

```
void
Height_toggled                  (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Dimeter_toggled                 (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Eco__Tec__toggled               (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Output_toggled                  (GtkToggleButton *togglebutton,
                                        gpointer        user_data);

void
Input_help_toggled              (GtkToggleButton *togglebutton,
                                        gpointer        user_data);
//support.h
//=====================================================
/*
  * DO NOT EDIT THIS FILE - it is generated by Glade.
  */

#ifdef HAVE_CONFIG_H
#  include <config.h>
#endif

#include <gtk/gtk.h>

/*
  * Standard gettext macros.
  */
```

```
#ifdef ENABLE_NLS
#  include <libintl.h>
#  undef _
#  define _(String) dgettext (PACKAGE, String)
#  define Q_(String) g_strip_context ((String), gettext (String))
#  ifdef gettext_noop
#    define N_(String) gettext_noop (String)
#  else
#    define N_(String) (String)
#  endif
#else
#  define textdomain(String) (String)
#  define gettext(String) (String)
#  define dgettext(Domain,Message) (Message)
#  define dcgettext(Domain,Message,Type) (Message)
#  define bindtextdomain(Domain,Directory) (Domain)
#  define _(String) (String)
#  define Q_(String) g_strip_context ((String), (String))
#  define N_(String) (String)
#endif


/*
  * Public Functions.
  */

/*
  * This function returns a widget in a component created by Glade.
  * Call it with the toplevel widget in the component (i.e. a window/dialog),
  * or alternatively any widget in the component, and the name of the widget
  * you want returned.
  */
GtkWidget*  lookup_widget              (GtkWidget       *widget,
                                        const gchar     *widget_name);
```

```
/* Use this function to set the directory containing installed pixmaps. */
void        add_pixmap_directory       (const gchar     *directory);


/*
 * Private Functions.
 */

/* This is used to create the pixmaps used in the interface. */
GtkWidget*  create_pixmap              (GtkWidget       *widget,
                                        const gchar     *filename);

/* This is used to create the pixbufs used in the interface. */
GdkPixbuf*  create_pixbuf              (const gchar     *filename);

/* This is used to set ATK action descriptions. */
void        glade_set_atk_action_description (AtkAction       *action,
                                              const gchar
*action_name,
                                              const gchar
*description);
```